Delinea

Secret Server

Administrator Guide

Version: 11.7.x

Publication Date: 4/28/2025

Secret Server Administrator Guide

Version: 11.7.x, Publication Date: 4/28/2025

Table of Contents

Secret Server Documentation

Introduction

Delinea Secret Server is an enterprise-grade password management solution designed to help organizations securely store, manage, and control access to privileged credentials. It aims to improve the security of sensitive data, reduce the risk of data breaches, and streamline the password management process.

Here are the key features of Delinea Secret Server:

- Secure Password Storage: Secret Server stores privileged credentials in an encrypted format, protecting sensitive information from unauthorized access.
- Access Control: Secret Server implements role-based access control, allowing administrators to set permissions and control who has access to sensitive information.
- Privilege Escalation Management: Secret Server integrates with Windows systems to provide privilege escalation management, helping to reduce the risk of data breaches.
- Auditing and Reporting: Secret Server provides detailed audit logs and reports, making it easier for organizations to track access to sensitive information and detect any unauthorized activity.
- Automated Password Management: Secret Server supports automated password management, helping to streamline the password management process and reduce the risk of manual errors.
- Multi-Factor Authentication: Secret Server supports multi-factor authentication, helping to improve the security of sensitive information.
- Integration with Other Tools: Secret Server integrates with a variety of other tools, including Active Directory, Microsoft Azure, and cloud-based applications, making it easier for organizations to manage their passwords and access controls.

This section of the Delinea documentation portal supports Secret Server.

Navigate using the dynamic table of contents on the left, the page contents on the right, or by entering a search term above. Many pages in this documentation have sub-pages. The container (parent) pages can have introductory text or simply a heading with no text. Please click the table of contents on the left to see any sub-pages it might have.

Documentation

You are at the home page of the current Delinea Document Portal for Secret Server. It contains:

- New material.
- Links to legacy knowledge bases article that have yet to be converted or archived. There are very few of these left.
- Links to legacy PDF documentation. These are also rare, and if you do find one, its target is very likely out of date.

Getting Started

- "Secret Server Business User Guide" on page 75 (for non-technical users)
- Getting Started Tutorial (for technical users)
- "Installation" on page 65
- "System Requirements for Secret Server" on page 86

Best Practices

- "Configuration Best Practices" on page 14
- "Discovery Best Practices" on page 550
- "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310 (PDF)
- "Security Hardening Guide" on page 1402

Download Secret Server

Product Downloads

Release Notes

"Secret Server Release Notes" on page 1566 (On-Premises and Cloud)

Delinea Blog

The Delinea blog offers cybersecurity insights and expertise from industry leaders. With a focus on zero trust, identity management, and privileged access, the blog provides practical advice and thought leadership to help organizations strengthen their security posture. As a trusted voice in cybersecurity, Delinea leverages the blog to empower customers embracing zero trust and demonstrate their commitment to innovation.

Video Tutorials

We offer a range of training videos on the Delinea training site. This is a subscription service. Please contact your account manager for details.

Developer Resources

"Developer Resources" on page 1473

Help

The Help section is a central resource designed to assist users in navigating and utilizing the product effectively. It includes essential documents such as Document Conventions, which clarify the format and terminology used throughout the documentation, and the Secret Server Glossary, providing definitions of key terms. Additionally, the Self-Help Resources provides other support, while the Technical Support section offers information on how to

contact support teams for further assistance. This section empowers users with the knowledge and tools needed for efficient problem-solving and system navigation.

Document Conventions

Capitalization

Technical writing is typically so awash in capitalization that it often denotes nothing and harms legibility. To counter that, in general, this document follows the IBM Style Guide rule:

"Do not capitalize the names of features and components unless they are sold separately or are trademarked."

More specifically, the only things capitalized in this document are:

- Company, person, country, geographic place, or organization names
- Official or trademarked products or services, unless they officially have atypical capitalization, for instance *iPod*.
- Acronyms and initializations
- When referring to any UI labels that are capitalized, including page names
- When the word begins a sentence or phrase

Code and Command Line Text

Variable text in literal typed-in text and command-line parameters follow these industry-wide standards:

- All code and command-line interface text appears in `monospaced` text.
- Required parameters appear in angle brackets: `ping <hostname>`
- Optional parameters appear in square brackets: `mkdir [-p] <dirname>`
- Repeated parameters are followed by ellipses: `cp <source1> [source2...] <dest>`
- Multiple choice items are separated by vertical bars and grouped by curly brackets: `netstat {-t|-u}`

Keyboard Shortcuts

- Keyboard keys are bolded and surrounded with square brackets: **[Enter]**
- Concurrent key presses are denoted with plus signs: **[Ctrl]+[Alt]+[Del]**
- Sequential key presses are denoted by commas: **[Page Down]**, **[Enter]**

Notes

There are two types of notes: *notes* and *important notes*.

A note contains tangential (an aside) or supplemental information (a tip or clarification).

An important note contains substantive information that should be heeded, or negative consequences can occur, involving frustration, wasted time, or even data loss.

Other Special Text

- Email addresses and URLs are usually denoted by a colored underline: support@delinea.com.
- When URLs are part of the instruction, as opposed to clickable link, they appear in monospaced text: Type `https://www.somewhere.com` or click https://www.somewhere.com.
- Cross-references to headings are hyperlinks: See Booting a Server.
- Document or article names (not sections) appear in italics: See the *Server Administration Guide*. They may or may not be hyperlinks.
- All file and folder paths appear in monospaced text: `app\bin\web_config.xml`
- File names by themselves do *not* appear in monospaced text: web_config.xml. If the file name contains spaces, the name is surrounded by quotation marks: "web config.xml".

Ending punctuation may be omitted for clarity when following typed-in text, including URLs.

Screen Components and Attentional Targets

- Mouse-click, keyboard, and other attentional targets (anything a looks for) are denoted by bold type: **OK** button or **Login** link.
- Attentional Targets and screen component names in system *responses* are not bolded: "The OK button appears" verses "Click the **OK** button."
- Names of screen components, such as tabs, buttons, and text boxes, are corrected for spelling and capitalization. The component type appears in lowercase.

Secret Server Glossary

Table: Terms and Definitions

Term	Definition	Description
2FA	Two-Factor Authentication	A security process in which a user provides two different authentication factors to verify their identity. Typically, this includes something they know (like a password) and something they have (such as a mobile device with a verification code).
AAD	Azure Active Directory	A cloud-based identity and access management service by Microsoft that allows users to sign in and access multiple resources such as Microsoft Office 365, the Azure portal, and thousands of other SaaS applications.

Term	Definition	Description
AC	Access Control	The process of granting or denying specific authorization to access data, functions, and other computer resources. It ensures that users access only what they are allowed to and prevents unauthorized users from accessing sensitive information.
Access Offline Secrets on Mobile		Role permission: Allows a user to cache their Secrets in the Secret Server mobile application for offline use. This permission does not automatically come with the Administrator role.
Access Request		The access request feature allows a secret to require approval prior to accessing the secret.
Access Request Workflows		See Workflows.
Account Lifecycle Manager (ALM)		Account Lifecycle Manager (ALM) controls the creation, management, and decommissioning of Active Directory Service Accounts running on your organization's network.
ACE	Access Control Entry	A part of an access control list that specifies the user or group permissions for a specific object, such as a file or directory.
ACL	Access Control List	A list of permissions attached to an object that specifies which users or system processes can access that object and what operations they can perform.
ACS	Access Control Server	A server used to authenticate and authorize users. It typically integrates with network access devices and identity databases to provide access control services.
Active Directory Account		Built-in secret template.
AD	Active Directory	A Microsoft technology used to manage computers and other devices on a network. It includes a range of services, such as LDAP directory services and Kerberos-based authentication.
Add Secret		Role permission: Allows a user to create new Secrets. The Add permission no longer include the role permission View Secret.

Term	Definition	Description
Add Secret (Folder Permission)		Allows the user to add a secret in that folder. Does not grant access to the added secret.
Add Secret Custom Audit		Role permission: Allows a user to make a custom audit entry when accessing a Secret using the web services API.
ADFS	Active Directory Federation Services	A software component developed by Microsoft that provides single sign-on (SSO) to authenticate a user in multiple web applications in a single session.
ADGUID	Active Directory Globally Unique Identifier	A unique identifier used within Microsoft's Active Directory for objects.
ADM	Administrator	
Administer Active Directory		Role permission: Allows a user to view domains, edit existing domains, delete domains, and add new domains. Also allows a user to force synchronization or set the synchronization interval.
Administer Automatic Export		Role permission: The user can do everything the other automatic export permissions allow and edit the automatic export configuration.
Administer Backup		Role permission: Allows a user to view and configure automated backups for Secret Server. Users with this role permission can change the backup path, disable backups, and set the backup schedule.
Administer Configuration		Role permission: Allows a user to view and edit general configuration options. For example, a user with this role permission can turn on "Force HTTPS/SSL" and disable "Allow Remember Me".
Administer Configuration Proxying		Role permission: Allows a user to view and edit SSH Proxy settings.
Administer Configuration SAML		Role permission: Allows a user to view and edit SAML integration settings on the Login tab of Configuration settings.

Term	Definition	Description
Administer Configuration Security		Role permission: Formerly "Administer Security Configuration," allows a user to view and edit security configuration options in Secret Server. Currently, these include enabling FIPS compliance mode and protecting the encryption key.
Administer Configuration Session Recording		Role permission: Allows a user to view and edit session recording settings on the Session Recording tab of Configuration settings.
Administer Configuration Two Factor		Role permission: Allows a user to change the configuration settings of the two factor authentication that are available for users logging into Secret Server.
Administer Configuration Unlimited Admin		Role permission: Formerly "Administer Unlimited Admin Configuration," allows a user to turn on Unlimited Admin Mode. When this mode is enabled, users with the "Unlimited Administrator" role permission can view and edit all Secrets in the system, regardless of permissions. Note that you can assign "Administer Unlimited Admin Configuration" to one user and "Unlimited Administrator" to another user. This would require one user to turn on the mode and another user to view and edit secrets.
Administer ConnectWise Integration		Role permission: Allows a user to view and edit configuration options for synchronizing with ConnectWise. This can be accessed through the "Folder Synchronization" link on the Administration page. Note that you need at least view access on the sync folder in order to set up or edit the ConnectWise integration.
Administer Create Application Accounts		Role permission: Formerly "Create Application Account", allows a user to create application user accounts to be used exclusively for accessing Secret Server via the API.
Administer Create Users		Role permission: Allows a user to create new local users in Secret Server, but not edit them once created.

Term	Definition	Description
Administer Custom Password Requirements		Role permission: Allows a user to view and edit custom password requirements that can be configured under the Security tab for individual Secrets.
Administer Data Retention		Role permission: Can manage audit data retention, such as editing and running now. This permission does not automatically come with the Administrator role.
Administer DevOps Secrets Vault Tenants		Role permission: Add, remove, and edit DSV tenants that automatically synchronize with Secret Server on a schedule.
Administer Disaster Recovery		Role permission: Allows a user to configure instances as data sources or replicas for Disaster Recovery. Also allows user to initiate or test Data Replication and view related logs and audits.
Administer Discovery		Role permission: Allows a user to view and import computers and accounts that are found by Discovery.
Administer Distributed Engine Configuration		Role permission: Allows a user to update the Distributed Engine configuration.
Administer DoubleLock Keys		Role permission: Allows a user to view, edit, create, and disable DoubleLock keys. A DoubleLock key acts as a separate encryption key to protect your most sensitive secrets. This option allows users to access and use the "DoubleLocks" link on the Administration page.
Administer Dual Control		Role permission: Allows a user to view, edit, create, and disable Dual Control settings for reports and recorded sessions.
Administer Event Subscriptions		Role permission: Allows a user to view, edit and create event subscriptions.
Administer Export		Role permission: Allows a user to view the export log. Also allows users to export Secrets to which they have access to a clear text, CSV file.

Term	Definition	Description
Administer Folders		Role permission: Allows a user to view, edit, create, move, and delete folders. Users still need the relevant view, edit, and owner permissions on the folders to perform these tasks.
Administer Groups		Role permission: Allows a user to view, edit, create, and disable groups. Also allows users to assign users to groups and remove users from groups.
Administer HSM		Role permission: Allows a user to change configuration or disable the use of a Hardware Security Module (HSM).
Administer Inbox		Role permission: Administer notification settings for the inbox.
Administer IP Addresses		Role permission: Allows a user to create, edit, and delete IP Address Ranges. These ranges are used to restrict certain users to specific IP Addresses.
Administer Jumpbox Route		Role permission: Allows a user to create, edit, or deactivate jump server routes.
Administer Key Management		Role permission: Allows a user to enable, change, or disable the Key Management (Secret Server Cloud only).
Administer Languages		Role permission: Allows a user to change the default language of Secret Server.
Administer Licenses		Role permission: Allows a user to view, edit, install, and delete licenses.
Administer Lists		Role permission: Add, remove, and modify lists and list contents in Admin > Lists.
Administer Metadata		Role permission: Manage metadata fields and sections added to secrets and users in Secret Server.
Administer Nodes		Role permission: Allows a user to view and edit server nodes and clustering settings.
Administer OpenID Connect		Role permission: Allows a user to manage OpenID connections.

Term	Definition	Description
Administer Password Requirements		Role permission: Allows a user to view and edit character sets and password requirements.
Administer Pipelines		Role permission: Allows a user to create, edit, and remove event pipelines and event pipeline policies.
Administer Platform Integration		Role permission: Allows a user to manage the Secret Server connection to the Delinea platform.
Administer Remote Password Changing		Role permission: Allows a user to turn Heartbeat and Remote Password Changing on and off globally. Also allows users to create new password changers and install password changing agents on remote machines.
Administer Reports		Role permission: Allows a user to view, edit, delete, and create reports. Also allows users to customize report categories.
Administer Role Assignment		Role permission: Allows a user to view which users and groups are assigned to which roles. Also allows users to assign users and groups to different roles.
Administer Role Permissions		Role permission: Allows a user to view, edit, create and delete roles. Also allows users to assign different permissions to each role.
Administer Scripts		Role permission: Allows a user to view, edit, and add PowerShell, SQL, and SSH scripts on the Scripts Administration page.
Administer Search Indexer		Role permission: Allows a user to view and edit search indexer options. These options control how searching in Secret Server works. For example, a user with this role permission could enable search indexing, which allows users to search on fields within a secret.
Administer Secret Policy		Role permission: Allows a user to create and edit Secret Policies.'
Administer Secret Templates		Role permission: Allows a user to view, edit, disable, and create Secret Templates.
Administer Security Analytics		Role permission: Allows a user to view and edit the settings for Privilege Behavior Analytics.

Term	Definition	Description
Administer Session Monitoring		Role permission: Allows a user to view and terminate active launcher sessions.
Administer SSH Menus		Role permission: Allows a user to edit and create SSH Menus, used in allowlisting commands that can be used on a SSH session.
Administer System Log		Role permission: Allows users to view and clear the System Log, which shows general diagnostics information for Secret Server.
Administer Teams		Role permission: Users can create, delete, and view all teams.
Administer Template Custom Columns		Role permission: Allows a user to enable the "Expose for Display" setting of a Secret template field to make it available for use in Dashboard custom columns.
Administer Users		Role permission: Allows a user to create, disable, and edit users in the system. This permission also allows a user to create and edit SDK/CLI rules.
Administer Workflows		Role permission: Allows users to manage workflows (advanced access management).
Administration Side Panel		The administration side panel and page is a control panel for administering Secret Server.
Administrator		Administrator is a default role that comes preconfigured with. Roles control access to features within. This role can be customized to have different permissions. In this guide, administrator (lowercase) is used when referring to users who manage the system and have control over global security and configuration settings. Note that administrators in do not automatically have access to all data stored in the system—access to data is still controlled by explicit permissions on that data.

Term	Definition	Description
ADMX	Administration XML	An ADMX file is a Group Policy Administrative Template file used with the Group Policy Management Console (GPMC) in Microsoft Windows operating systems. ".admx" is the file extension. The file format is XML-based and replaces the older ADM format, which was used in earlier versions of Windows.
ADS	Advanced Directory Services Integration.	Allows Secret Server to integrate with Active Directory for user authentication and management.
ADSI	Active Directory Service Interfaces	A set of COM interfaces used by Windows to interact with and manage Active Directory.
Advanced Import		Role permission: Allows a user to import Secrets from an XML file. Users with the this permission can import groups, folders, site connectors, sites, and secret templates, without having to create a secret. Users must have the Secret Server permissions needed for the objects listed in the XML.
Advanced Session Recording		Advanced Session Recording (ASR) is a licensed feature of Secret Server that adds capabilities to those offered by basic session recording. You install the Advanced Session Recording Agent (ASRA), which uses the Remote Desktop Protocol, on any client machine where you want more information from the sessions recorded.
Advanced Session Recording Agent		The session recording software on a user's computer that performs ASR.
AES	Advanced Encryption Standard	A symmetric encryption algorithm widely used across the globe.
AES128	128-bit AES	
AES192	192-bit AES	
AES256	256-bit AES	
AIX	IBM's Unix operating system	

Term	Definition	Description
AJAX	Asynchronous JavaScript and XML	A set of web development techniques using various technologies to create asynchronous web applications.
All Secrets		All Secrets is a master table of the secrets stored on . It is a one-stop, searchable location for examining the status and properties of secrets. It is a supplement to, not a replacement for, the secret folder tree. It lists and you can sort by secret template, heartbeat status, sync status, machine, access date, username, and much more. You can customize which characteristics are displayed.
Allow Access Challenge		Role permission: Allows a user be challenged by Privileged Behavior Analytics if their behavior deviates from their normal behavior and meets certain requirements set by Privileged Behavior Analytics. Administrators do not have this permission by default.
Allow List Secret Access for Assigning Policy		Role permission: Allows users with list access to a secret to assign policies. Users need the view permission if they do not have this one.
ALM	Account Lifecycle Manager	See Account Lifecyle Manager.
Amazon IAM Console Password		Built-in secret template.
Amazon IAM Key		Built-in secret template.
AMQP	Advanced Message Queuing Protocol	A protocol that enables client applications to communicate with middleware brokers.
ANSI	American National Standards Institute	A private non-profit organization that oversees the development of voluntary consensus standards for products, services, processes, systems, and personnel in the U.S.
APAC	Asia Pacific region	A geographic region comprising East Asia, South Asia, Southeast Asia, and Oceania.
API	Application Programming Interface	A set of protocols and tools for building software and applications.

Term	Definition	Description
APM	Application Performance Monitoring	The practice of monitoring software application performance to ensure quality and detect issues.
Application Dashboard		The main page for searching and viewing secrets. You access it by clicking the Dashboard menu item.
Approve via Duo Push		Role permission: Allow a user to approve access requests via Duo push notifications. Administrators do not have this permission by default.
ARCFOUR	Alleged RC4 cipher	A stream cipher that was used in TLS prior to TLS 1.2.
ARGV	Argument Vector	Represents command-line arguments that are passed to a program.
ARN	Amazon Resource Name	A unique identifier for AWS resources, allowing for easy identification and interaction within AWS.
AS400	IBM mid-range server brand	Refers to IBM's range of mid-sized servers, now known as IBM i.
ASA	Adaptive Security Appliance	A security device from Cisco that combines firewall, antivirus, intrusion prevention, and virtual private network (VPN) capabilities.
ASCII	American Standard Code for Information Interchange	A character encoding standard for electronic communication.
ASHX		A web file extension used with Microsoft's ASP.NET | Open-source web framework for .NET for handling HTTP requests.
ASP	Advanced Server Pages	Microsoft's server-side script engine for dynamically generated web pages.
ASPNET	ASP.NET | Open-source web framework for .NET	A framework for building dynamic web sites, applications, and services.
ASR	Advanced Session Recording	A feature that records user sessions for monitoring or compliance purposes.
ASRA	Advanced Session Recording Agent	A component responsible for capturing the data needed for session recording.

Term	Definition	Description
Assign Pipelines		Role permission: Allows the user to assign an event pipeline policy to secret policies, or folders.
Assign Secret Policy		Role permission: Allows a user to assign Secret Policies to folders and secrets.
Automatic Secret Export		This feature allows you to automatically export secrets on a schedule to an external location in an encrypted, password-protected archive.
Automatic Sudo or Su Privilege Elevation		A convenience feature that eliminates the need to manually enter a su or sudo command's password when using a proxied SSH session to a Unix or Linux server. When a user manually types a su or sudo command with a valid secret ID, the SSH proxy automatically provides the username and password to use. The user does not need to know either.
AWS	Amazon Web Services	A subsidiary of Amazon providing on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis.
Azure AD Account		Built-in secret template.
Bank Account		Built-in secret template.
Basic Session Recording		Basic session recording is a licensed feature in Secret Server. It relies on the protocol handler configured on client machines through Secret Server's launcher. Using the launcher, Secret Server captures second-by-second screenshots on the client machine during a user's recorded session. These images of the user's screen are compiled into a video that can be downloaded and played back for auditing and security purposes. Activity recorded in the session is based on screen changes only.
BLOWFISH		An encryption method where the same key is used for both encrypting and decrypting the data.
BMC	Business Machine Code	Machine code or instructions used to operate business machinery such as mainframe computers.

Term	Definition	Description
Browse Reports		Role permission: The "Browse Reports" role allows access to reports restricted by permissions. Permissions are configurable at the category and report levels and share a similar inheritance model to secrets and folders. You can define users or groups with "view" or "edit" permissions for each category or report.
BSD	Berkeley Software Distribution	A Unix operating system derivative developed and distributed by the Computer Systems Research Group at the University of California, Berkeley.
Bypass Direct API Authentication Restriction		Role permission: Allows users to ignore the PreventDirectApiAuthentication advanced setting and log in via the API with a non-application account.
Bypass SAML Login		Role permission: Allows a user to login with local account without using SAML.
CA	Certificate Authority	An entity that issues digital certificates used to verify the identity of the certificate holder and provide the public key necessary to enable secure communications.
CAC	Common Access Card	A smart card used primarily by the United States Department of Defense for identification and secure access.
CAL	Certificate Authority List	A list of trusted certificate authorities that a browser or operating system will trust for secure communications.
CALs	Client Access Licenses	A license that grants a user or device the right to access services, such as RDS, from a server running the Windows Server operating system.
CAPTCHA	Completely Automated Public Turing test to tell Computers and Humans Apart	A challenge-response test used in computing to determine whether the user is human.
CAST	CAST-128 encryption algorithm	A symmetric encryption algorithm used to encrypt and decrypt data, utilizing a 128-bit key.

Term	Definition	Description
CAST128	128-bit key CAST encryption algorithm	A variant of the CAST-128 algorithm that specifically utilizes a 128-bit encryption key.
CBC	Cipher Block Chaining	A block cipher mode that provides confidentiality by combining each plaintext block with the previous ciphertext block before encryption.
CC	Common Criteria	An international standard for evaluating the security of information technology products.
CD	Compact Disc	A digital optical disc used to store data, including music and data files.
CDATA	Character Data	Special notations used in XML to represent characters that may be confused with markup elements.
CDN	Content Delivery Network	A system of distributed servers that deliver pages and other web content to users based on their geographical location.
CEF	Common Event Format	A standardized text file format for log management and interoperability among security tools.
CER	Certificate	A file format and extension that holds digital certificates
CERT	Computer Emergency Response Team	An expert group that handles computer security incidents.
CFG	Configuration file	A file used to configure the operations of a computer program.
CGI	Common Gateway Interface	A standard for interfacing external applications with servers to produce dynamic web pages.
CHACHA20	ChaCha20 stream cipher	A modern symmetric encryption algorithm designed for speed and security.
Checkout Hook		In addition to changing the password on check in, secret owners can also specify administrator-created PowerShell scripts, called hooks, to run before or after checkout and check in. These are accessed from the Hooks tab of the secret, which only shows if checkout is enabled and PowerShell scripts have been created by an admin.

Term	Definition	Description
CHGUSRPRF	Change User Profile	A command used to change the values specified in a user profile on an IBMi, iSeries, or AS400 system
CI	Continuous Integration	A software development practice where developers regularly merge their code changes into a central repository, followed by automated building and testing.
CID	Column Identifier	A unique identifier for a column in a database or other data structure.
CIDR	Classless Inter-Domain Routing	A method for allocating IP addresses and routing Internet Protocol packets.
CIFS	Common Internet File System	A protocol that allows programs to make requests for files and services on remote computers on the Internet.
CIS	Center for Internet Security	A nonprofit organization that provides cybersecurity tools, benchmarks, and guidelines.
Cisco Account (SSH)		Built-in secret template.
Cisco Account (Telnet)		Built-in secret template.
Cisco Enable Secret (SSH)		Built-in secret template.
Cisco Enable Secret (Telnet)		Built-in secret template.
Cisco VPN Connection		Built-in secret template.
CISO	Chief Information Security Officer	A senior-level executive responsible for an organization's information and data security.
CKM	Cryptographic Key Management	The process of managing cryptographic keys for a cryptosystem, including generation, use, storage, exchange, and replacement.
CLI	Command Line Interface	Allows managing systems such as Secret Server via command-line scripts and tools.

Term	Definition	Description
Cloud Suite		Cloud Suite is a part of a broader portfolio of identity and access management solutions, designed to extend privileged access management (PAM) and identity services to cloud environments. This enables organizations to secure access to cloud resources and applications through the same identity platform that they use for on-premises resources.
CLR	Common Language Runtime	A managed execution environment that is part of Microsoft's .NET framework.
CLSID	Class Identifier	A GUID that represents a specific class within the .NET framework.
CM	Configuration Management	The process of maintaining the consistency and integrity of a system or product throughout its lifecycle.
CM	Connection Manager	See Connection Manager.
CMAK	Connection Manager Administration Kit	A tool used to manage network connections in Windows.
CMDLINE	Command Line	An interface that allows users to interact with software by typing commands.
CMK	Customer Master Key	A key used in Amazon Web Services Key Management Service to encrypt and decrypt data.
CMVP	Cryptographic Module Validation Program	A program by the National Institute of Standards and Technology to validate cryptographic modules to Federal Information Processing Standards.
CN	Common Name	An attribute used in the subject field of a certificate to identify the entity that the certificate represents.
CNG	Cryptography API: Next Generation	An application programming interface for cryptography, provided by Microsoft.
COM	Component Object Model	A binary-interface standard for software components to communicate.
COM+ Dependency Scanner		The COM+ Dependency Scanner allows for an Active Directory domain discovery source to locate COM+ Applications running on machines on the domain that are being run by Domain Accounts.

Term	Definition	Description
Combination Lock		Built-in secret template.
Common Criteria		The Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408), known as "Common Criteria," is an international standard for certifying security of computer systems, networks, and application software. Certification ensures that claims about the security attributes of the evaluated product have been independently verified in the specified, evaluated configuration and environment. The certification does not validate any security claims when the product is used outside of that evaluated configuration or environment.
Company Policy Login Banner		A login banner for usage agreements and conditions to be visible when users log into Secret Server
Configuration Overview Page		The Configuration Overview page (also called the "configuration preview") is a dynamic, one-stop locator for Secret Server configurations.
Connection Manager (CM)		Connection Manager provides secure connections to remote servers using RDP and SSH, allowing IT teams to launch ad-hoc connections to manage sessions with remote resources. Management of multiple active sessions is easy. You can store and organize connections by adding them to your favorites and import any folder structure or connections used in other tools for a single management hub. It includes remote connectivity tools closely integrated with Secret Server. It permits technical staff to quickly access resources using the convenience of a familiar, rich desktop interface while maintaining all the safeguards and workflows included with Secret Server.
Contact		Built-in secret template.
Copy Secret		Role permission: Allows a user to copy secrets when that user also has Own Secret role permission.
CORS	Cross-Origin Resource Sharing	A security feature implemented in web browsers to control the interactions between web pages from different domains.

Term	Definition	Description
CPAN	Comprehensive Perl Archive Network	A repository of over 250,000 software modules and accompanying documentation for the Perl programming language.
CPIC	Common Policy Implementation Criteria	Criteria used to assess the alignment of policies and procedures with common or accepted standards.
CPU	Central Processing Unit	The primary unit of a computer that performs most of the processing.
CQRS	Command and Query Responsibility Segregation	A pattern in software architecture where the data modification command responsibilities are separated from the data reading query responsibilities.
Create Root Folders		Role permission: Allows a user to create new folders at the root level of the folder structure.
Credit Card		Built-in secret template.
CRL	Certificate Revocation List	A list of digital certificates that have been revoked by the issuing certificate authority.
CRM	Customer Relationship Management	An approach to manage a company's interactions with current and potential customers using data analysis.
CRQ	Change Request	A formal proposal for an alteration to some product or system.
CRT	Certificate	File extension
CSA	Cloud Security Alliance. An industry group focused on cloud security best practices that Secret Server aligns with.	An industry group focused on cloud security best practices, including alignment with tools like Secret Server.
CSR	Certificate Signing Request	Used by systems like Secret Server to obtain SSL certificates from a certificate authority.
CSS	Cascading Style Sheets	A style sheet language used for describing the look and formatting of a document written in HTML.
CSV	Comma Separated Values	A file format that stores tabular data in plain text, with columns separated by commas.

Term	Definition	Description
CTL	Certificate Trust List	A list of trusted certificates used by Windows to determine if a certificate issued by a particular certificate authority is to be trusted.
Custom Password-Exclusion Dictionary		A list of words that you do not want users to choose as part of a password, for example, your company name. The dictionary becomes an option when creating or editing a password requirement object. Those, in turn, appear as options when creating a secret template. Finally, when a secret is created based on that template, the words in the dictionary are not allowed when creating a password (the "weak" warning appears).
CVE	Common Vulnerabilities and Exposures	A dictionary of publicly known information security vulnerabilities and exposures.
CVSS	Common Vulnerability Scoring System	A standard used to classify and rate the severity of security vulnerabilities in software.
Ciphertext		The result of encryption performed on plaintext using a cipher algorithm.
DAT	Data	File extension.
DB	Database	A structured collection of data stored in a computer, often in a tabular form, and managed using software to facilitate rapid search and retrieval.
DBA	Database Administrator	A person responsible for maintaining and optimizing a database, and ensuring its availability, performance, and security.
DBMS	Database Management System	Software that is used to create, manage, and manipulate databases.
DBO	Database Owner	A role in database systems that has complete control over the database, including permissions, schema modification, and data manipulation.
DC	Domain Controller	In a network, it is a server that responds to security authentication requests and maintains the security policy and user account information.

Term	Definition	Description
DCOM	Distributed Component Object Model	An extension of the Component Object Model (COM) that allows COM objects to communicate across network boundaries.
DCS	Dynamic Credentials	A feature that automatically updates or rotates credentials based on configurable rules to enhance security.
DE	Distributed Engine	An engine that spreads tasks across multiple servers or nodes to distribute the workload and improve performance.
Deactivate Secret		Role permission: Allows a user to mark secrets as deactivated.
Delete Secrets from Reports		Role permission: Allows a user to run the delete Secrets action from a report.
Delinea Mobile App		The Delinea Mobile app provides MFA verification for the Delinea Platform as well as portable access to secrets managed in Secret Server.
Delinea Platform (DP)		The Delinea Platform seamlessly extends privileged access management across your company's hybrid multi-cloud infrastructure, with adaptive controls that help IT and cybersecurity teams to rapidly meet compliance and reduce risk.
Dependency (Secret)		Secret dependencies are items that rely on the username, password, or SSH private key stored in the secret. By adding them to the Dependencies tab, they are automatically updated when the secret's password is changed, ensuring they are up to date with the account on which they depend.

Term	Definition	Description
Dependency Group (Secret)		By default, all dependencies are updated in the order listed. There are cases where you may want to split out different sets of dependencies into separate groups. Typically, this is because a single service account may run services across different segregated networks that can communicate with the domain but not each other and have different distributed engine sites assigned. In this case you can create two dependency groups and assign them to different distributed engine sites to solve connectivity issues.
DES	Data Encryption Standard	A symmetric encryption algorithm that was widely used but is now largely obsolete due to its vulnerability to brute-force attacks.
DevOps	Development and Operations	A set of practices that involve the collaboration and communication of both software developers and IT professionals to automate the process of software delivery and infrastructure changes.
DevOps Secrets Vault (DSV)		Delinea's DevOps Secrets Vault is a high velocity vault that centralizes secrets management, enforces access, and provides automated logging trails. This cloud-based solution is platform agnostic and designed to replace hard-coded credentials in applications, micro-services, DevOps tools, and robotic process automation. This vault ensures IT, DevOps and Security teams the speed and agility needed to stay competitive without sacrificing security. DevOps Secrets Vault is deployed as an API-as-a Service.
DevOps Secrets Vault Client Credentials		Built-in secret template.
DH	Diffie-Hellman key exchange algorithm	A method of securely exchanging cryptographic keys over a public channel, often used in secure communications protocols.
DHE	Ephemeral Diffie-Hellman key exchange	A variation of the Diffie-Hellman key exchange that uses temporary or "ephemeral" keys for each session, increasing security.

Term	Definition	Description
DIM	Dimension	In computing and data science, this term often refers to a particular aspect or feature of data, used to provide some context or analysis. In databases, it could refer to an attribute or set of attributes in a data set used to provide some form of classification.
Directory Services		Directory services are components of network operating systems that map the names of network resources to their network addresses. Their shared information infrastructure locates, manages, and organizes network resources, which can include volumes, folders, files, users, groups, devices, and much more. Active Directory is Secret Server's native directory service.
Discovery		Discovery is the process where scans an environment to find accounts and associated resources called dependencies. Once accounts are found, you can use them to create associated new secrets in . Users with the "administer discovery" role permission can either manually import accounts or can create an automated process, called a discovery rule, to do so. Using discovery does not stop users from manually creating their own secrets.
Discovery Command Set		An SSH script that runs on Unix machines and produces a specific set of output to be consumed in a discovery source flow.
Discovery Scan Template		Defines an object and what properties the object contains. For example, a computer account has a name, machine, and domain. Think of a scan template as an interface that describes an object.
Discovery Scanner		A discovery scanner is a component of a discovery source that collects information during a discovery. There are four general types of scanners, called scan templates. Defines how to take that information and runs code to produce collection outputs. Scanners can be system out-of-the-box code that runs natively in the system or completely custom scripts that can do anything.
Discovery Script		A script for a discovery scanner.

Term	Definition	Description
Discovery Secret Search Filters		Certain scanners and import rules can use a filter that uses the name of the machine to find or use an associated Secret. For example, you may have a pattern of naming the local account on a machine including the machine name. A secret search filter allows you to find secrets using the name of the current machine in the pattern to find the matching secret.
Discovery Source		A discovery source is a named collective, ordered system that conducts discovery. There are five broad types: Active Directory, Amazon Web Services, Unix, VMware ESX\ESXi, and Google Cloud Platform.
Discovery Source Flow		A collection of scanners that work in a common pipe and filter architecture where each scanner inputs a certain type of item and then outputs a different type of item. For example, a scanner takes an input of a host IP range and outputs multiple computers that can then be consumed by another scanner which can input computer information and output computer accounts.
Distributed Engine		For smaller enterprises, Secret Server performs all functions on the Web server it is installed on. It is also scalable for large enterprises and scenarios demanding higher performance. We use remote distributed engines to accomplish this. You route high-demand processing and traffic operations through one or more of these to enhance 's capacity. For example, distributed engines can synchronize and authenticate for Active Directory. They can also perform remote password changing, heartbeat, discovery and more, all controlled by a single installation.
distributedengine		A Windows service that does a DE's actual work, such as password changing, heartbeat, discovery, and more. Each engine belongs to a site.
DKIM	DomainKeys Identified Mail	An email authentication technique that allows the receiver to check that an email was actually sent by the domain it claims to have been sent by and that it hasn't been changed in transit.

Term	Definition	Description
DLL	Dynamic-link library	A collection of small programs or routines designed to perform specific tasks and can be dynamically loaded into running programs.
DMZ	Demilitarized Zone	A physical or logical subnetwork that contains and exposes an organization's external-facing services to an untrusted network, usually the internet.
DN	Distinguished Name	A string used in the X.500 standard to uniquely identify an entity within a directory service.
DNS	Domain Name System	A system for converting human-friendly domain names into IP addresses.
DOM	Document Object Model	A programming interface that represents HTML and XML documents as a tree structure where each node is an object representing a part of the document.
DOS	Disk Operating System	A family of disk operating systems predominantly composed of MS-DOS and a rebranded version under the name IBM PC DOS.
DoubleLock		Secret Server's doublelock is a feature that provides an additional security layer by protecting secret data using asymmetric encryption (a public/private key pair) where the private key is a human-generated password. This feature is independent of regular permissions, Secret Server login access, or physical access to the machine running Secret Server. A shortcut way of thinking about doublelocks is as an extra password for secrets that is held by a set group of users. In addition, both the password and the group of users are reusable for other secrets.
DoubleLock Object		A named object that is associated with one or more secrets and one or more users (via password objects). DoubleLock objects, or simply doublelocks, point to secrets (what can be accessed) and doublelock password objects (who can access it).

Term	Definition	Description
DoubleLock Password Object		An encrypted password that is associated with one user. The same QuantumLock password object, or simply doublelock password, is used for all doublelocks to which a user has access. Once a user is assigned to a doublelock, that user has access to any secret using that doublelock, using a single password. A doublelock password has nothing to do with the user's Secret Server access password.
Download Automatic Export		Role permission: The user can view all of the automatic export tabs and download exports from cloud storage (cloud customers only).
Download Hash		Download hash codes or hashes are used to verify the integrity and authenticity of downloaded software. Hash codes are unique mathematical values that are generated based on the content of the software file and can be used to confirm that the downloaded file matches the original version and has not been tampered with. By comparing the hash code of the downloaded file with the hash code provided by the software developer, you can ensure that the software you downloaded is genuine and has not been corrupted during the download process.
DP	Delinea Platform	See Delinea Platform.
DPAPI	Data Protection API	A Windows API used by applications like Secret Server to encrypt sensitive data at rest.
DR	Disaster Recovery	The process, policies, and procedures related to preparing for and recovering from a serious negative event affecting information systems.
DRAC	Dell Remote Access Controller	A hardware and software solution for remote systems management.
DRP	Disaster Recovery Plan	A documented process or set of procedures that helps in the recovery or protection of a particular IT infrastructure in the event of a disaster.

Term	Definition	Description
DS	Directory Service	Software systems that store, organize, and provide access to directory information in order to reduce duplication of information.
DSA	Digital Signature Algorithm	A public-key algorithm used for digital signatures.
DSC	Desired State Configuration	A management platform in PowerShell that enables you to manage your IT and development infrastructure declaratively.
DSEE	Directory Server Enterprise Edition	An enterprise directory service product from Oracle, offering robustness and scalability.
DSOH	Directory Server On-Demand Hold	A feature in directory services that temporarily prevents specific changes from replicating to other servers.
DSS	Data Security Standard	Security standards designed to secure data through a network.
DSV	DevOps Secrets Vault	See DevOps Secrets Vault.
DUO	Duo Security	A company that provides multi-factor authentication services.
DWORD	Double Word (data type)	A data type that usually consists of 32 bits in a computer's memory.
EA	Early Availability	Refers to a beta version of software that is made available for testing before the official release.
EA	Enterprise Architecture	A conceptual blueprint that defines the structure and operation of an organization.
EAP	Extensible Authentication Protocol	A framework for transport of authentication protocols.
EAU	Endpoint Authentication	The process of authenticating devices that connect to a network.
EBS	Elastic Block Store (AWS)	Provides block storage volumes for use with Amazon EC2 instances.

Term	Definition	Description
EC2	Elastic Compute Cloud (AWS)	An AWS service that provides resizable computing capacity in the cloud.
ECC	Elliptic Curve Cryptography	An asymmetric key encryption technology.
ECDH	Elliptic Curve Diffie-Hellman	A version of the Diffie-Hellman protocol using elliptic curve cryptography.
ECDSA	Elliptic Curve Digital Signature Algorithm	A variant of the Digital Signature Algorithm (DSA) which uses elliptic curve cryptography.
Edit (Folder Permission)		Allows the user to create new folders in that folder, which forces the "Inherit Permissions from Parent" permission on the new folder, move secrets into that folder, and add new secrets into that folder.
Edit Secret		Role permission: Allows a user to edit secrets. Note that they still require the "Edit" or "Owner" permissions on the individual secrets they are editing.
EFS	Encrypting File System (Windows)	A feature of Windows that allows files to be transparently encrypted to protect confidential data.
EID	Endpoint Identifier	A unique ID that helps in identifying network endpoints.
EMEA	Europe, Middle East, and Africa	A commonly used regional designation.
Engine	Distributed Engine	An engine that spreads tasks across multiple servers or nodes to distribute the workload and improve performance.
ENU	English	One of the standardized set of codes used to represent the English language, such as "en" or "en-US".
ENV	Environment	In computing, refers to a specific set of hardware, software, and settings used for specific tasks.
EP	End Point	The point at which a data flow ends, like a destination IP address and port number in a network.
EPMD	Erlang Port Mapper Daemon	A service in Erlang programming for distributing and managing nodes.

Term	Definition	Description
Erase Secret		Role permission: Allows a user to permanently erase (as opposed to deactivate, which is reversible) a secret.
ERP	Enterprise Resource Planning	A software system that helps organizations manage and integrate their core business processes. It provides a centralized platform for various departments within an organization to collect, store, manage, and interpret data from different business activities
ESM	Enterprise Security Manager	A security module in SAP systems that helps in managing different security aspects.
EST	Enrollment over Secure Transport protocol	A protocol used for secure certificate enrollment.
ESX	Elastic Sky X	VMware project. Short for VMware ESXi hypervisor, a virtualization platform.
Event Pipelines		Event pipelines (EPs) are a named group of triggers, filters, and tasks to manage events and responses to them. Event pipelines themselves can be grouped into EP policies. The Secret Server EP system is essentially a flexible instruction set builder and manager for controlling events and responses.
Event Subscriptions		Event subscriptions trigger notifications of defined events within the system. These notifications are sent to the inbox and from there can be sent externally via email or Slack.
EVT	Event	An event log file extension for Windows.
EWSR	Enable Web Services Reporting	An option to activate reporting capabilities via web services.
EX	Export Policy	A set of guidelines or rules that govern how data or services can be transferred or accessed outside a particular environment.
Expire Secrets from Reports		Role permission: Allows a user to expire Secrets listed in a report.'

Term	Definition	Description
Extensible Discovery		Extensible discovery lets you extend the already powerful scanning abilities of Secret Server by creating custom scanners that run PowerShell. You can use either built-in or custom scanners and templates at each step of the discovery process in extensible discovery. If the built-in discovery sources, scanners, or input and output template, cannot you meet your needs, you can use PowerShell scripts to perform any part of discovery. Doing so requires that you define your own input and output templates and scanners and then add them to a new or existing discovery source.
FAU	Federal Agency Unit	A unit of the federal government responsible for performing government functions, ranging from intelligence and defense to public policy and regulations.
FIA	Financial Institutions Advisories	Advisories issued by the U.S. Treasury targeted at financial institutions to notify them of regulatory changes or security concerns.
FIDO2	Fast Identity Online 2	A set of standards that enable simpler and more secure user authentication experiences across many types of platforms and devices.
Field Slug Name		A field slug name in Secret Server is a unique human-readable identifier for a data field in a Secret Server template. The field slug name is available for integrating with third-party applications via API calls. Slug names are programmatically available for API calls but are not visible to template users (secret creators). Instead, they are displayed as references in secret templates.
FIPS	Federal Information Processing Standard	A set of U.S. government standards that define how various types of information are to be encoded. Secret Server's support for FIPS refers to its compatibility with FIPS 140-2 validated cryptographic modules.

Term	Definition	Description
Folders		Secret folders allow you to create containers of secrets based on your needs. They help organize your customers, computers, regions, and branch offices, to name a few. Folders can be nested within other folders to create sub-categories for each set of classifications. Secrets can be assigned to these folders and sub-folders. Folders allow you to customize permissions at the folder level, and all secrets within can inherit the folder's permissions. Setting permissions at the folder level ensures future secrets placed in that folder have the same permissions, simplifying management across users and groups.
Force Check In		Role permission: Allows a user to force a secret that is checked out by another user to be checked in.
FPS	Frames Per Second	A unit that measures display device performance or video/animation quality, indicating how many individual frames are displayed per second.
FPSE	Format Preserving Strong Encryption	Encryption that transforms plaintext into ciphertext of the same format and length, typically used in financial systems.
FQDN	Fully Qualified Domain Name	The complete domain name for a specific computer, or host, on the internet, often consisting of a host name and a domain name, like Example Domain .
FTA	Functional Test Associate	Typically a role or position involved in the functional testing of software or systems to ensure they meet specified requirements.
FTP	File Transfer Protocol	A standard network protocol used to transfer computer files from one host to another over a TCP-based network, such as the internet or an intranet.
GA	General Availability	The stage in a software's lifecycle where it is considered fully tested and reliable, often the first version available to the general public.
GCM	Galois/Counter Mode (cryptography)	A mode of block cipher operation providing both data authenticity (integrity) and confidentiality. Commonly used in cryptography.

Term	Definition	Description
GCM	Google Cloud Messaging	A messaging solution for sending information from servers to client apps. It is now replaced by Firebase Cloud Messaging (FCM).
GCP	Google Cloud Platform	A suite of cloud computing services that run on the same infrastructure that Google uses internally for its end-user products.
GDPR	General Data Protection Regulation	A regulation enforced by the EU to protect citizens' personal data and privacy, affecting businesses worldwide.
Generic Discovery Credentials		Built-in secret template.
Geo Replication		Server geo-replication is a technique used to distribute data across multiple geographic locations for the purpose of data redundancy, high availability, and faster access. It is commonly used in the context of database servers, web servers, and other types of infrastructure that require fault tolerance and quick recovery from hardware failures or other types of incidents.
Getting Started Tutorial		A introductory guide to Secret Server for technical users.
Global Banner		A multipurpose global banner for all users and used for maintenance, security, or policy notifications.
Google IAM Service Account Key		Built-in secret template.
GPO	Group Policy Object	A feature of the Microsoft Windows NT family of operating systems that allows administrators to implement specific configurations for users and computers within an Active Directory environment.
GPU	Graphics Processing Unit	Specialized hardware designed to accelerate rendering of images and videos to be output to a computer's screen.

Term	Definition	Description
Group		A collection of users managed together for administrative convenience. Users within a group inherit the permissions and roles attributed to that group, making it easier to manage security settings. Groups can also be synchronized with external services like Active Directory.
GSS	Generic Security Services	A software layer that provides a standardized interface for secure communication, often used to implement Kerberos mechanisms.
GUI	Graphical User Interface	A type of user interface that allows users to interact with computer systems through graphical elements like buttons and windows, as opposed to text-based interfaces.
GUID	Globally Unique Identifier	A unique reference number used as an identifier in computer software, ensuring that each object has a unique ID.
HA	High Availability	A system design approach aimed at ensuring a high level of operational uptime. It often involves redundant or failover components.
Healthcare		Built-in secret template.
Heartbeat		Secret Server's heartbeat feature allows secrets to have their entered credentials automatically tested for accuracy at a given interval. Using heartbeat on secrets ensures those credentials are up-to-date and can alert administrators if the credentials are changed outside of Secret Server. Heartbeat helps manage secrets and prevent them from being out of sync.
HMAC	Hash-based Message Authentication Code	A specific type of message authentication code that uses cryptographic hash functions and a secret key for data integrity and authentication.
HP	Hewlett-Packard	An American multinational information technology company that produces hardware and software solutions.

Term	Definition	Description
HPE	Hewlett Packard Enterprise	Hewlett Packard Enterprise Servers are robust and scalable servers designed to meet the complex demands of modern business computing, offering advanced features for virtualization, high-performance computing, and cloud-based environments.
HP iLO Account (SSH)		A built-in secret template.
HSM	Hardware Security Module	Physical computing devices that safeguard digital keys and perform cryptographic functions.
HSTS	HTTP Strict Transport Security	A web security policy mechanism that helps to protect websites against man-in-the-middle attacks by enforcing secure (HTTPS) connections.
HTML	Hypertext Markup Language	The standard markup language for documents designed to be displayed in web browsers.
HTTP	Hypertext Transfer Protocol	The fundamental protocol used for transferring data over the web.
HTTPAPI	HTTP Application Programming Interface	A set of API calls for HTTP, allowing for programmatic interaction with web services.
HTTPS	HTTP Secure (SSL/TLS encryption)	An extension of HTTP, secured with SSL/TLS encryption.
IAM	Identity and Access Management	A framework for business processes that facilitates the management of electronic identities, allowing the right individuals to access the right resources at the right times for the right reasons.
IBM	International Business Machines Corporation	An American multinational technology and consulting company.
IBM iSeries Mainframe		Built-in secret template.
ICD	Interface Control Document	A document that describes the interface to a system or subsystem, providing details for ensuring compatibility.
ICT	Information and Communications Technology	An umbrella term that includes any communication device or application, encompassing radio, television, phones, computer and network hardware, etc.

Term	Definition	Description
ID	Identifier	A unique value used to identify a record or entity within a database or system.
IDEA	International Data Encryption Algorithm	A symmetric key block cipher that was once considered very secure but has since been superseded by newer algorithms.
Identity Bridge		A legacy Active Directory management product.
IDP	Identity Provider	A system that creates, maintains, and manages identity information and provides authentication services.
IDS	Intrusion Detection System	
IE	Internet Explorer	A discontinued web browser developed by Microsoft.
IIS	Internet Information Services	A web server from Microsoft used to host websites and other content on the web.
iLO	Integrated Lights Out	An embedded server management technology exclusive to Hewlett Packard Enterprise, which allows for remote control of HPE servers, providing powerful management capabilities irrespective of the server's operating status.
Inbox		The Inbox page shows notifications such as event subscription alerts, access requests and approvals, and other configuration alerts in a single interface. In addition to viewing notifications in the inbox, you can configure the inbox to forward them via email or Slack, subject to numerous configurable criteria. You can also customize the format of the email messages.
INTG	Integrations	Generally refers to the process of bringing together different computing systems and software applications physically or functionally.
IO	Input/Output	The collection of interfaces that different functional units of an information processing system use to communicate with each other.
IOS	Internetwork Operating System	The software used on a majority of Cisco Systems routers and switches.

Term	Definition	Description
iOS	iPhone Operating System	Apple smartphone OS.
IoT	Internet of Things	Refers to a network of physical devices that are embedded with sensors, software, and other technologies to collect and exchange data with other devices and systems over the internet. Secret Server can manage credentials for IoT devices.
IP	Internet Protocol	The principal communications protocol for relaying datagrams across network boundaries in the internet.
IPSec	Internet Protocol Security	A suite of protocols for securing internet protocol (IP) communications by authenticating and encrypting each IP packet in a data stream—a VPN tunnel.
IPV	IPv4	The fourth version in the development of the internet protocol (IP) and routes most traffic on the internet.
ISAPI	Internet Server API	An API developed by Microsoft that allows you to extend the functionality of an IIS web server.
ISE	Identity Services Engine	A security policy management and control platform from Cisco Systems.
ISO	International Organization for Standardization	An international standard-setting body composed of representatives from various national standards organizations.
ITSM	Information Technology Service Management	A set of policies, processes, and procedures for managing IT services.
IWA	Integrated Windows Authentication	An authentication method to securely store usernames and passwords, commonly used in intranet environments.
JAR	Java Archive	A package file format that aggregates many files into one, typically used to store Java classes.
JIRA	Jira	An issue and project tracking software developed by Atlassian.
JS	JavaScript	A high-level, interpreted programming language used for client-side web development.

Term	Definition	Description
JSON	JavaScript Object Notation	A lightweight data-interchange format that's easy for humans to read and write and easy for machines to parse and generate.
JWT	JSON Web Token	An open standard for securely transmitting information between parties as a JSON object.
KB	Knowledge Base	A database used for storing information, commonly in the context of technical support.
KBA	Knowledge Base Article	An individual document in a knowledge base, typically addressing a single issue or topic.
KBA	Knowledge-Based Authentication	A security measure that requires the user to answer a question only they would know.
KEM	Key Encryption Mechanism	An algorithm used to securely encapsulate (encrypt) cryptographic keys.
KEX	Key Exchange algorithm	Algorithms used to securely exchange cryptographic keys between parties.
KMS	Key Management Service	A service that manages cryptographic keys for encryption. It uses envelope encryption where the encryption key changes each time you make a request for a key.
Kyber-1024		A KEM specifically designed to resist quantum computing attacks.
LAN	Local Area Network	A network that connects computers within a limited area, like a home or office.
LCID	Locale Identifier	A unique identifier assigned to geographical or cultural regions for formatting purposes.
LDAP	Lightweight Directory Access Protocol	A protocol for accessing and maintaining a directory service, such as Microsoft's Active Directory.
LDAPS	LDAP Secure	LDAP over SSL/TLS, a secure version of LDAP.
LDP	Lightweight Directory Protocol	Usually a typo referring to LDAP.

Term	Definition	Description
LDS	Lightweight Directory Services	A subset of LDAP services that's lighter and easier to manage.
LM	LAN Manager	An outdated protocol suite by Microsoft used to provide file and print sharing services.
LPT	Line PrinTer (port)	Refers to the printer port used in older computer systems.
LSA	Local Security Authority	A Windows component responsible for enforcing security policies.
LTS	Long Term Support (version)	Software versions that are supported for a longer period than standard versions.
LWP	Libwww-perl	A Perl library for web-related activities.
Maintenance Mode		Maintenance mode allows you to temporarily prevent users from changing roles, secrets, or secret-related data such as dependencies, templates, and password requirements.
Master Encryption Key		The main key used in a cryptographic system, often used to derive additional keys. When Secret Server is first installed, a unique random AES256 Master Encryption Key (MEK) is generated and saved in a file, encryption.config. The MEK protects anything sensitive in Secret Server that is not associated with a specific secret, as well as each secret's unique AES256 key when an HSM is not used.
Master Encryption Key Rotation		For added security, you can rotate the MEK, re-encrypting protected data with the new key.
MD2-5	Message Digest 2-5	Cryptographic hash functions.
MEK	Master Encryption Key	See Master Encryption Key.
MemoryMQ	Memory Message Queue	A legacy built-in service developed by Delinea—replaced with RabbitMQ for production systems.
MFA	Multi-Factor Authentication	A security system that requires multiple methods of authentication.

Term	Definition	Description
Microsoft SQL Server		A relational database management system developed by Microsoft.
MMC	Microsoft Management Console	A framework for hosting administrative tools on Windows.
MOF	Managed Object Format	A language for describing instances of CIM (Common Information Model) classes.
MP4	MPEG-4 video file format	A digital multimedia container format most commonly used to store video and audio.
MPEG	Moving Picture Experts Group	A working group responsible for the development of video and audio encoding standards.
MQ	Message Queue	Middleware from IBM that helps to seamlessly connect different components of a business application.
MS	MicroSoft	An abbreviation for Microsoft Corporation.
MSADC	Microsoft Active Directory Connector	A way to integrate Microsoft Active Directory with other services.
MSDTC	Microsoft Distributed Transaction Coordinator	A component of Microsoft Windows that is responsible for coordinating transactions.
MSI	Microsoft Installer	A software component used for installing, maintaining, and removing software.
MySql Account		Built-in secret template.
NARTAC	NATO Communications and Information Agency	An agency responsible for NATO's IT and communication needs.
NAS	Network-Attached Storage	A dedicated file storage system that provides storage space over a network.
NAT	Network Address Translation	A technique of remapping an IP address space into another.
NATO	North Atlantic Treaty Organization	A military alliance of 30 countries.

Term	Definition	Description
NCO	NATO Consultation	Likely refers to the process or instance of consulting or collaboration within the context of NATO.
NCRYPT	Ncrypt	A library in Windows for handling cryptography functions contained in nncrypt.dll. It is an essential component of the operating system and is used by various programs to ensure the security of sensitive information. This DLL file provides encryption and decryption operations, helping to protect data from unauthorized access or tampering.
NDP	Network Device Enrollment Service	A Cisco service for secure distribution of certificates.
NET	Microsoft .NET Framework	A software framework developed by Microsoft.
NETBIOS	Network Basic Input/Output System	An API that augments the DOS API and provides network services.
NG	Next Generation	Generally refers to the next iteration or version of a product, technology, or methodology.
NIST	National Institute of Standards and Technology	An agency that develops and promotes measurement standards.
NNTP	Network News Transfer Protocol	An Internet application protocol for the distribution, retrieval, and posting of news articles.
NPL	National Physical Laboratory	The national measurement standards laboratory for the United Kingdom.
NSA	Network Security Appliance	The SonicWall Network Security Appliance (NSA) series is a line of advanced firewalls designed by SonicWall, targeted primarily at medium to large-sized businesses and distributed enterprise environments.
NSA	National Security Agency	A U.S. intelligence agency responsible for signal intelligence and information assurance.
NT	New Technology	Usually refers to a family of operating systems by Microsoft, starting with Windows NT.

Term	Definition	Description
NTC	National Transportation Commission	An organization that regulates transportation.
NTFS	New Technology File System	A file system used by Windows NT and its successors.
NTLM	NT LAN Manager	An authentication protocol used in various Microsoft network protocol implementations.
NTP	Network Time Protocol	A protocol used to synchronize computer clock times over a network.
OAEP	Optimal Asymmetric Encryption Padding	A padding scheme often used in RSA encryption.
OATH	Open Authentication	Standards for multi-factor authentication (MFA).
OAUTH	Open Authorization framework for API authentication	A framework for token-based API authentication.
Object Metadata		Object metadata allows you to store extended information on several Secret Server objects including users, groups, folders, dates, or secrets via the user interface or REST API. You can store most data types, including strings, Boolean values, numbers, dates, and users. You can combine this metadata into sections containing named fields of your defined types.
OCS	Office Communications Server	A Microsoft instant messaging platform.
ODAC	Oracle Data Access Components	Software components for connecting to Oracle databases.
ODBC	Open Database Connectivity	A standard API for accessing database management systems.
OEM	Original Equipment Manufacturer	A company that produces parts and equipment that may be marketed by another manufacturer.
OIDC	OpenID Connect	An identity protocol that is layered on top of the OAuth 2.0 protocol.

Term	Definition	Description
OKTA	Okta	A service providing identity management solutions.
OLDAP	OpenLDAP	An open-source implementation of the Lightweight Directory Access Protocol.
OOB	Out of Box	
OpenLDAP Account		Built-in secret template.
ORA	Oracle database	A database management system from Oracle Corporation.
Oracle Account		Built-in secret template.
Oracle Account (TCPS)		Built-in secret template.
Oracle Account (Template Ver 2)		Built-in secret template.
Oracle Account (Walletless)		Built-in secret template.
ORG	Organization	A group of people working together towards common goals.
OS	Operating System	Software that controls the computer hardware and provides services for computer programs.
OSX	Operating System X	A desktop operating system from Apple.
OTP	One-Time Password	A password that is valid for only one login session.
OU	Organizational Unit	A subdivision within an Active Directory into which you can place users, groups, computers, and other OUs.
Own Group		Role permission: Allows a user to be an owner of a group. This permission is in the default Group Owner role, which is automatically assigned when that user is set as owner of a group.

Term	Definition	Description
Own Secret		Role permission: Formerly "Share Secret", allows a user to share secrets with other users. Also allows users to perform more advanced tasks on secrets of which they are "Owners", such as configuring expiration schedules, configuring the web launcher, converting secret template, and copying secrets (when a user also have the Copy Secret role permission.)
Own User		Role permission: Allows the user to become a user owner, used to configure specific users without the Administer Users permission.
Owner (Folder Permission)		Allows the user to create new folders in that folder without forcing inheritance, move the folder, delete the folder, rename the folder, and change the permissions and inheritance settings on the folder.
PAM	Privileged Access Management	A solution that helps organizations restrict privileged access within an environment.
PAP	Password Authentication Protocol	A simple, clear-text authentication scheme.
Password		Built-in secret template.
PBA	Privileged Behavior Analytics	See Privileged Behavior Analytics.
PBKDF2	Password-Based Key Derivation Function 2	A key derivation function with a sliding computational cost.
PC	Personal Computer	A computer designed for use by one person at a time.
PCI	Payment Card Industry	A set of security standards designed to ensure that all companies that accept, process, store or transmit credit card information maintain a secure environment.
PCI DSS	Payment Card Industry Data Security Standard	A set of security standards for merchants and organizations that handle branded credit cards.
PDF	Portable Document Format	A file format used to present and exchange documents reliably.

Term	Definition	Description
Personal Folder		In Secret Server, a personal folder is a folder that one (and only one) individual has owner access to. No user can modify sharing permissions on these folders. User's can add subfolders to their personal folder. The purpose of this folder is to allow a user to securely store work-related secrets that other users do not require access to. Note that when in break-the-glass mode, an unlimited admin can access a user's personal folder in order to recover secrets if needed.
Personal Folders		Role permission: Allows a user to have personal folder when the global personal folders configuration options is enabled.
Personally Identifiable Information		Any data that could potentially identify a specific individual.
PFX	PKCS #12	A file format for storing server cryptographic keys.
PII	Personally Identifiable Information	Any data that could potentially identify a specific individual.
PIN	Personal Identification Number	A numerical code used for authentication.
Pin		Built-in secret template.
PIV	Personal Identity Verification	A standard for smart cards used for identification.
PKCS	Public Key Cryptography Standards	A set of cryptographic standards.
PKI	Public Key Infrastructure	A framework that manages digital keys and certificates.
PM	Project Manager	An individual responsible for planning, executing, and closing projects.
PM	Privilege Manager	See Privilege Manager.
PNG	Portable Network Graphics	A raster graphics file format that supports lossless data compression.

Term	Definition	Description
POC	Proof of Concept	A demonstration to verify that certain concepts or theories have the potential for real-world application.
PORT	Network port	A hardware or software interface for transferring data.
POSIX	Portable Operating System Interface standard	A family of standards specified by the IEEE for maintaining compatibility between operating systems.
POST	Power On Self Test	A diagnostic process that occurs when you start the computer.
PostgreSQL	An open-source database management system	A free and open-source relational database management system.
PQ	Post Quantum	An encryption or encapsulation algorithm that is resistant to quantum computing attacks.
Privilege Manager (PM)		Privilege Manager is an endpoint least privilege and application control solution for Windows, macOS, and Unix/Linux, capable of supporting enterprises and fast-growing organizations at scale. Mitigate malware and modern security threats from exploiting applications by removing local administrative rights from endpoints. The two major components are Local Security and Application Control.
Privilege Manager Administrator		Role permission: Allows the user to have the "Administrator" role for Privilege Manager, giving full access to the system.
Privilege Manager Helpdesk User		Role permission: Allows the user to have the "Help Desk" role for Privilege Manager, giving full access to approve or deny escalation requests.
Privilege Manager MacOS Admin		Role permission: Allows the user to have the MacOS "Administrator" role for Privilege Manager, giving full access to the system.
Privilege Manager Unix/Linux Admin		Role permission: Allows the user to have management permissions to Unix/Linux policies and machines.

Term	Definition	Description
Privilege Manager User		Role permission: Allows the user to have the "User" role for Privilege Manager, giving read and write permissions to most items, but not rights to modify security permissions. Administrators do not have this permission by default.
Privilege Manager Windows Administrator		Role permission: Allows the user to have the Windows "Administrator" role for Privilege Manager, giving full access to the system.
Privileged Behavior Analytics (PBA)		PBA works with Secret Server to improve enterprise system security by helping to visualize, detect, interrupt, and announce threatening activity and behavior.
Product License Key		Built-in secret template.
Protocol Handler		An application on an end-user's machine. It enables communication between Secret Server and that client machine. It also provides the files needed by launchers.
PS	PowerShell	A task automation and configuration management framework from Microsoft.
PSE	Personal Security Environment	A user-specific environment with customized security settings.
PUB400		Public access folders in an IBM i environment.
Public Worker		A role that utilizes a public bus for communication and is not intended for executing business application logic.
PuTTY	Popular SSH and Telnet Client (literally)	A free and open-source terminal emulator that supports various network protocols.
QA	Quality Assurance	The process of verifying or determining whether products meet or exceed customer expectations.
QR	Quick Response	A type of matrix barcode commonly used for storing URLs or other information.
QRTZ	Quartz enterprise scheduler	An open-source job-scheduling library.

Term	Definition	Description
RabbitMQ (RMQ)		RabbitMQ is an important component of Secret Server's on-premises environment, providing a robust framework for queuing messages between Secret Server and its Distributed Engines. RabbitMQ is an enterprise-ready software package that provides reliability and clustering functionality superior to other applications. RabbitMQ helper is Delinea's implementation of RabbitMQ.
RabbitMQ Durable Exchange		In RabbitMQ, an exchange is a routing mechanism that takes messages from producers and pushes them to queues based on certain rules, known as bindings. An exchange can be configured to be "durable," meaning it will survive server restarts.
RabbitMQ Helper		See RabbitMQ.
RACF	Resource Access Control Facility	IBM mainframe security module.
RADIUS	Remote Authentication Dial-In User Service	A networking protocol that provides centralized Authentication, Authorization, and Accounting.
RAM	Random Access Memory	A type of computer memory that can be read and changed in any order.
RAS	Remote Access Service	A service for remote computer access.
RBAC	Role-Based Access Control	A system for managing permissions based on roles within the organization.
RBS	Role Based Security	An approach to security that assigns permissions based on roles within the organization.
RCE	Remote Code Execution	The ability to execute code on a remote system.
RCP	Remote Copy Protocol	A protocol used for copying files over a network.
RD	Remote Desktop	A technology that allows users to connect to a remote computer.
RDBMS	Relational Database Management System	A database management system based on the relational model.

Term	Definition	Description
RDP	Remote Desktop Protocol	See Remote Desktop Protocol
Remote Desktop Protocol		A Microsoft protocol for remote control of computers.
RDPWin		The primary executable for SSPH.
Remote Desktop Services		Remote control services (using RDP) provided by a dedicated server or servers.
RDS	Remote Desktop Services	See Remote Desktop Services.
ReBAC	Relationship-Based Access Control	A type of access control based on relationships between entities.
REG	Windows Registry	A database to store settings and options for the Microsoft Windows operating system.
Remote Password Changing		A Secret Server feature that can automatically change passwords on various platforms including Windows, databases, and network appliances. Remote Password Changing (RPC) allows secrets to automatically update a corresponding remote account. You can set secrets for automatic expiration, followed by automatic strong password generation and a remote password update to keep the subject accounts synchronized with Secret Server.
REQS	Requirements	Specifications that must be satisfied by a system or component.
REST	Representational State Transfer	An architectural style for distributed hypermedia systems.
REST API	RESTful Application Programming Interface	An API using REST architecture. Used in Secret Server.
RFC	Request for Comments	A publication from the IETF and the Internet Society, the principal technical development and standards-setting bodies for the Internet.
RHEL	Red Hat Enterprise Linux	An enterprise-level Linux operating system.

Term	Definition	Description
RIPEMD	Race Integrity Primitives Evaluation Message Digest	A cryptographic hash function.
RIPEMD160	160-bit RIPEMD	A 160-bit cryptographic hash function.
RMQ	RabbitMQ	See RabbitMQ.
Role		Every user and group must be assigned to a role. uses role-based access control to provide very granular system access. ships with three roles: Administrator, User, and Read-Only User. Each role contains a set of permissions to match the job function of users with that role. See the Role Permissions List for details.
Rotate Encryption Keys		Role permission: Allows a user to start a process that rotates the Secret encryption keys.
RPC	Remote Password Changing	See Remote Password Changing.
RPC	Remote Procedure Call	A protocol for executing code on a remote server.
RPO	Recovery Point Objective	Maximum acceptable data loss in the event of a failure.
RSA	Rivest-Shamir-Adleman	A public key encryption technology.
RTO	Recovery Time Objective	The maximum amount of time for recovering data after a disaster.
RU	Request Unit	A unit of measure for system resources.
Run Automatic Export		Role permission: The user can view all of the automatic export tabs and run the export manually by clicking the Run Export button.
Run Disaster Recovery Data Replication		Role permission: Allows user to initiate or test Data Replication.

Term	Definition	Description
Run Scripts		Role permission: Separates privileges in script management. Holders of the "View Scripts" role permission cannot execute test runs of scripts, and this permission must be assigned to perform this task.Administer Scripts remains unchanged and allows view, edit, and run permissions.
S3	Simple Storage Service	AWS object storage.
SA	Systems Administrator	Person responsible for managing systems.
SALT		Random data added during password hashing.
SAM	Security Accounts Manager	Windows component for managing security accounts.
SAML	Security Assertion Markup Language	An open standard used by Secret Server to support SSO.
SAN	Storage Area Network	A network designed for storing data.
SAP	Systems Applications and Products	An ERP system and company.
SAP Account		Built-in secret template.
SAP SNC Account		Built-in secret template.
SAPNCO	SAP NetWeaver Component	A component of SAP NetWeaver, which is a software stack and technology platform developed by SAP SE. It serves as the technical foundation for many of SAP's enterprise applications
SAPSNC	SAP SNC	Cryptographic library for SAP.
SBS	Small Business Server	Windows server edition tailored for small businesses.
SCIM	System for Cross-domain Identity Management	Protocol for managing identities across domains.
SCP	Secure Copy Protocol	SSH file transfer protocol.
SDK	Software Development Kit	Toolkit for software development.

Term	Definition	Description
SDLC	Software Development Life Cycle	The phases involved in software development.
SEC	Security and Exchange Commission	U.S. government agency for regulating securities.
Secret		A piece of information that is stored and managed within is referred to as a secret. Secrets are derived from secret templates. Typical secrets include, but are not limited to, privileged passwords on routers, servers, applications, and devices. Files can also be stored in secrets, allowing for storage of private key files, SSL certificates, license keys, network documentation, Microsoft Word or Excel documents and more.
Secret Checkout		The Secret Server checkout feature forces accountability on secrets by granting exclusive access to a single user. If a secret is configured for check out, a user can then access it. If Change Password on Check In is turned on, after check in, Secret Server automatically forces a password change on the remote machine. No other user can access a secret while it is checked out, except unlimited administrators. This guarantees that if the remote machine is accessed using the secret, the user who had it checked out was the only one with proper credentials at that time.
Secret Key Rotation		Secret key rotation is a somewhat similar process to RPC by which the encryption key, used for securing secret data, is changed and that secret data is re-encrypted. Each secret receives a new, unique AES-256 key. Secret key rotation can be used to meet compliance requirements that mandate encryption keys be changed on a regular basis.
Secret Launch		Role permission: Dictates whether or not a user can launch a secret. Previously, a user could launch a secret if their user's role had the "View Secret" permission. As of Version 11.5, a user needs this permission to launch. A user will also need the "Secret Launch Remote Access (Platform)" permission to be able to launch a Remote Session with (RAS)

Term	Definition	Description
Secret Launch Remote Access (Platform)		Role permission: Dictates whether or not a user can launch a secret. Previously, a user could launch a secret if their user's role had the "View Secret" permission. As of Version 11.5, a user needs this permission to launch a remote session with RAS.
Secret Launcher		A secret launcher launches applications on end-user machines and automatically logs on using credentials stored in Secret Server. In general, there are three types of launchers: RDP, SSH, and Custom. This provides a convenient method to open RDP and PuTTY connections, but it also circumvents users needing to know their passwords–a user can still gain access to a needed machine but it is not required to view or copy the password out of Secret Server. A Web launcher automatically logs into websites using the client's browser.
Secret Navigation Slideout		The Secret Navigation Slideout is a set of useful links to secret. Its tab appears on the right side of all top-level pages.
Secret Server		Delinea Secret Server is an enterprise-grade password management solution designed to help organizations securely store, manage, and control access to privileged credentials. It aims to improve the security of sensitive data, reduce the risk of data breaches, and streamline the password management process.
Secret Server Cloud Quick Start		A quick-start guide intended for business users of Secret Server Cloud.
Secret Server Migration Tool		A migration utility for importing secrets from other applications

Term	Definition	Description
Secret Server Mobile		Through the Secret Server Mobile application, users can connect from a mobile device to a Secret Server instance to view, manage, and use secrets stored there. The mobile application interface is similar to the Secret Server interface, which makes it easy for users to navigate to find secrets and secret folders. The mobile application offers useful functionality including multi-factor authentication, biometric authentication, autofill, online and offline caching, and advanced secret workflows.
Secret Template		Secret templates are used to create secrets and allow customization of the format and content of secrets to meet company needs and standards. Examples include: local administrator account, SQL Server account, Oracle account, credit card and Web password. Templates can contain passwords, usernames, notes, uploaded files, and drop-down list values. New secret templates can be created, and all existing templates can be modified.
Security Alarm Code		Built-in secret template.
Security Hardening Guide		A document that outlines security hardening for securing your Secret Server instance, whether it be installed on a single server or in a multi-clustered environment.
SEIM	Security Information and Event Management	System for security management.
SERPENT128	128-bit Serpent	A 128-bit encryption algorithm.
SERPENT192	192-bit Serpent	A 192-bit encryption algorithm.
SERPENT256	256-bit Serpent	A 256-bit encryption algorithm.

Term	Definition	Description
Server Suite		Server Suite is a comprehensive security solution designed to manage, secure, and audit both user and privileged account access across a wide range of servers, including Windows, Linux, and UNIX systems. The suite is particularly focused on minimizing the risk associated with privileged access, which can be a significant security vulnerability if not managed correctly.
Session Connector		Normally, Secret Server requires installing additional software such as Connection Manager or Secret Server Protocol Handler (SSPH) on the end-user computers to launch secrets, such as RDP, SSH, or custom, and optionally record the session. With Secret Server Session Connector (SSSC) installed on a Remote Desktop Services (RDS) server, anyone who can download and launch a standard Remote Desktop Protocol (RDP) shortcut file can have the same experience. The RDS server itself runs a special SSPH for RDS–SSPH (RDS) as a remote app to record the sessions, so end-users do not need to install any additional software.
Session Recording Auditor		Role permission: Grants access to the session recording of a secret to a user with at least "List Access" permission on the secret. Administrators do not have this permission by default.
SFTP	SSH File Transfer Protocol	Secure file transfer protocol.
SHA	Secure Hash Algorithm	A set of cryptographic hash functions.
SHA256	256-bit SHA-2 hash function	A 256-bit cryptographic hash function.
SHA384	384-bit SHA-2 hash function	A 384-bit cryptographic hash function.
SHA512	512-bit SHA-2 hash function	A 512-bit cryptographic hash function.
SID	Security Identifier	An identifier for security principles in Windows.

Term	Definition	Description
SIEM	Security Information and Event Management	System for security management.
Site (Distributed Engine)		A bucket of work items for a particular network area. Each engine is assigned to a single site, but each site can include multiple engines, significantly increasing throughput.
Site (Secret Server Cloud)		In Secret Server Cloud, the site serves as the designated location for storing secrets. Any operations necessitating a Distributed Engine (DE) will utilize the site specified, along with the DEs associated with that site.
Site Connector		A Windows service that holds the work items for a number of sites. The site connector can be either RabbitMQ or MemoryMQ (a built-in service developed by Delinea). Each site can only be assigned to a single site connector, but you can have multiple site connectors running on separate machines, each storing work items for multiple sites. Those sites, in turn, distribute the work items among multiple engines. The ability to add new Site Connectors, Sites, and Engines as needed makes Distributed Engine a highly-scalable solution.
SLA	Service Level Agreement	A contract for service quality.
SLO	Service Level Objective	Objective measurements for service quality.
SMB	Server Message Block	A network file-sharing protocol.
SMB Fallback		To maximize compatibility across versions of Windows when a heartbeat fails Secret Server makes a second attempt to use the Secret via SMB when Use SMB heartbeat fallback is checked. When Use SMB heartbeat fallback is not selected this second attempt will not be made.
SMS	Short Message Service	Text messaging.
SMTP	Simple Mail Transfer Protocol	Protocol for sending emails.

Term	Definition	Description
SNC	SAP NetWeaver AS for ABAP - Security Network Communications	Security component for SAP.
SNMP	Simple Network Management Protocol	Protocol for network management.
SOAP	Simple Object Access Protocol	Protocol for exchanging structured information.
SOC	Security Operations Center. Secret Server provides audit logs for SOCs monitoring security.	Center for monitoring and responding to security incidents.
Social Security Number		Built-in secret template.
SonicWall NSA Web Admin Account		Built-in secret template.
SonicWall NSA Web Local User Account		Built-in secret template.
SP	Service Pack	A set of updates for software.
SPN	Service Principal Name	Identifier for service instances.
SQL	Structured Query Language	Language for database queries.
SQL Server Account		Built-in secret template.
SRV	Server	Server.
SS	Secret Server	Server for managing secrets.
Secret Server Cloud	Secret Server Console	Console for managing Secret Server.
SSDE	Secret Server DevOps Edition	DevOps edition of Secret Server.
SSH	Secure SHell	Protocol for secure remote access.

Term	Definition	Description
SSH CA	SSH Certificate Authority	Secret Server can act as a CA for signing SSH certificates.
SSH Jumpbox Route		A series of regular Linux servers, accessible from the Internet, that is a gateway to other Linux machines on a private network using the SSH protocol. This topic and its subtopics address discuss using jumpbox routes.
SSH Key		Built-in secret template.
SSH Key Rotation		SSH Key Rotation allows you to manage your Unix account private keys and passphrases as well as their passwords. With key rotation, whenever the password is changed on the secret (manually, during a scheduled auto-change, or when checking in a secret that changes the password on check-in), the public/private key pair will be regenerated and the private key encrypted using a new passphrase. The public key will then be updated on the Unix machine referenced on the secret.
SSHD	SSH Daemon (server process)	Server process for SSH.
SSL	Secure Socket Layer	Protocol for secure communications.
SSMS	SQL Server Management Studio	Management studio for SQL Server.
SSO	Single Sign-On	Technology for allowing single sign-on.
SSP	Secret Server Provider	Provider for Secret Server.
SSPH	Secret Server Protocol Handler	See Protocol Handler.
SSPH (RDS)	Secret Server Protocol Handler, RDS Version	A special SSPH for use with SSSC that enables optional keystroke recording.
SSSC	Secret Server Session Connector	See Session Connector.
STIG	Security Technical Implementation Guide	Guidelines for securing systems.

Term	Definition	Description
SUDO	Substitute User DO command	Command for temporary superuser access.
SUPM	SAP User and Profile Management	User and profile management for SAP.
SUSE	SUSE Linux distribution	A distribution of Linux.
Sybase Account		Built-in secret template.
SYS	System	System.
SYSDBA	Oracle database superuser role	Superuser role for Oracle database.
T1	Thycotic One	See Thycotic One.
TCP	Transmission Control Protocol	Protocol for reliable data transmission.
TCP445	TCP port 445	Used by SMB for file sharing.
TCPS	TCP over TLS	TCP traffic over TLS encryption.
TDC	Tableau Data Connection file	File type used by Tableau for data connections.
TDE	Transparent Data Encryption. SQL feature used by Secret Server to encrypt database contents.	SQL feature used by Secret Server for database encryption.
Teams		See User Teams.
Thycotic One		A legacy access management product.
TID	Thread ID	Identifier for a thread in computing.
TLS	Transport Layer Security. Cryptographic protocol used by Secret Server for secure communications.	Cryptographic protocol for secure communication.

Term	Definition	Description
TMS	Ticketing Management System	System for managing tickets in IT support.
TNS	Transparent Network Substrate (Oracle networking)	Oracle networking component.
TOAD	Toad database admin tool	Database administration tool.
TOE	Target of Evaluation	Subject of a security evaluation.
TOP	TakeOut Point	Point where data is extracted.
TOTP	Time-Based One-Time Password	A type of one-time password that is time-based.
TPM	Trusted Platform Module hardware chip	Hardware chip for secure computing.
TRP	Trusted Realm Participant	Participant in a trusted security realm.
TS	Terminal Services	Services to enable multiple user access to Windows OS.
TSF	TOE Security Functionality	Security features in a TOE.
TSG	Technical Support Group	Group that provides technical support.
TSS	TOE Summary Specification	Summary of TOE security features.
TTL	Time to live	Time limit for data packets in a network.
TTLS	Tunneled Transport Layer Security	TLS within another protocol.
UAC	User Account Control	Windows feature for user permission control.
UDP	User Datagram Protocol	Protocol for simple, connectionless data transfer.
UI	User Interface	Visual interface of a software application.

Term	Definition	Description
UMAC	Universal Hashing Message Authentication Code	A type of message authentication code (MAC) calculated choosing a random hash function from a class of hash functions. The resulting digest or fingerprint is then encrypted to hide the identity of the hash function used.
UNC	Universal Naming Convention (Windows network paths)	Windows network path naming.
UNIX	Unix operating system	An operating system.
Unix Account (Privileged Account SSH Key Rotation - No Password)	Built-in secret template.	
Unix Account (Privileged Account SSH Key Rotation)	Built-in secret template.	
Unix Account (SSH Key Rotation - No Password)	Built-in secret template.	
Unix Account (SSH Key Rotation)		Built-in secret template.
Unix Account (SSH)		Built-in secret template.
Unix Account (Telnet)		Built-in secret template.
Unix Root Account (SSH)		Built-in secret template.
Unlimited Administration Mode		An emergency, break-the-glass mode that gives administrators access to all content within the system, regardless of explicit permissions. Access to unlimited administration mode is controlled using role permissions.
Unlimited Administrator		Role permission: Allows a user to view and edit all secrets in the system, regardless of permissions, when Unlimited Admin Mode is on. Note that another user with the "Administer Unlimited Admin Configuration" role permission would still need to turn this mode on.

Term	Definition	Description
Unrestricted by Teams		Role permission: Users can view all users, groups, and sites, regardless of team affiliation. Essentially, teams do not exist for the users with this permission, and the Teams page is not available to them. The default user role has this permission.
UPN	User Principal Name	User identifier in a Windows domain.
URI	Uniform Resource Identifier	A unique sequence of characters that identifying a logical or physical web resource.
URL	Uniform Resource Locator	A type of URI that identifies a web address, including the protocol.
USB	Universal Serial Bus	A standard for connecting devices.
User		Users are 's representation of people—one person per user. Each user has a unique username, as well as other attributes. Users are assigned to groups, and roles are assigned to them, either directly or via groups.
User Audit Expire Secrets		Role permission: Allows a user to view the "User Audit" report, which shows all secrets that have been accessed by a particular user in a specified date range. Also allows the user to force expiration on all these secrets, which would make Secret Server automatically change the password.
User Group		Secret Server allows administrators to manage users through user groups. Users can belong to different groups and receive the sharing permissions, as well as roles, attributed to those groups. This setup simplifies the management of the permissions and roles that can be assigned to a user. Additionally, groups can be synchronized with Active Directory to further simplify management.

Term	Definition	Description
User Teams		With Secret Server teams, administrators can create special groups called teams to restrict what users can see. A team bundles users and groups to assign them the same rules as to what other users and sites are visible to them. For example, a managed service provider could isolate their customers from seeing other customer's user accounts or a large company could "firewall" their users by department. Site visibility can also be restricted by teams.
UTC	Coordinated Universal Time	Standard time.
UTCNOW	Current time in UTC	Current time in Coordinated Universal Time.
UTF	Unicode Transformation Format	Encoding for Unicode characters.
UTF8	Unicode Transformation Format 8-bit	8-bit Unicode encoding.
UTILS	Utilities	Utility tools or programs.
UX	User Experience	Overall experience of a user using a product.
View (Folder Permission)		Allows the user to see the folder and secrets in that folder that are inheriting permissions from their folder.
View About		Role permission: Allows a user to view the "About" page from the Help menu, which links to external resources such as Technical Support and the Delinea blog.
View Active Directory		Role permission: Allows a user to view, but not edit, the Active Directory settings in the system.
View Advanced Dashboard		Role permission: Allows a user to view advanced dashboard. Without this permission, users will only be able to view basic dashboard.
View Advanced Secret Options		Role permission: Allows a user to view the Remote Password Changing, Security, and Dependency tabs on a Secret they have access to.

Term	Definition	Description
View Automatic Export		Role permission: The user can view all of the automatic export tabs.
View Backup		Role permission: Allows a user to view, but not edit, the automated backup settings.
View Configuration		Role permission: Allows a user to view, but not edit, general configuration settings.
View Configuration Proxying		Role permission: Allows a user to view, but not edit, SSH Proxy settings.
View Configuration SAML		Role permission: Allows a user to view SAML integration settings on the Login tab of Configuration settings.
View Configuration Security		Role permission: Formerly "View Security Configuration," allows a user to view the security configuration of Secret Server.
View Configuration Session Recording		Role permission: Allows a user to view session recording settings on the Session Recording tab of Configuration settings.
View Configuration Two Factor		Role permission: Allows a user to view the configuration settings of the two factor authentication that are available for users logging into Secret Server.
View Configuration Unlimited Admin		Role permission: Formerly "View Unlimited Admin Configuration," allows a user to view the Unlimited Admin Mode configuration. Also allows a user to view the Unlimited Admin Mode audit log.
View ConnectWise Integration		Role permission: Allows a user to view, but not edit, the ConnectWise integration settings.
View Data Retention		Role permission: Can view retained audit data. This permission does not automatically come with the Administrator role.
View Deleted Secrets		Role permission: Allows a user to view Secrets that have been deleted in the system.
View DevOps Secrets Vault Tenants		Role permission: View (not edit) the DSV tenants set to synchronize with Secret Server.

Term	Definition	Description
View Disaster Recovery		Role permission: Allows a user to view configuration, logs and audits for Disaster Recovery.
View Discovery		Role permission: Allows a user to view, but not edit, computers and accounts that are found by Discovery.
View Distributed Engine Configuration		Role permission: Allows a user to view the Distributed Engine configuration.
View DoubleLock Keys		Role permission: Allows a user to view which DoubleLock keys exist in the system.
View Dual Control		Role permission: Allows a user to view configured Dual Control settings for reports and Secret sessions.
View Enterprise Objects		Role permission: Allows a user to view user and secret metadata.
View Event Subscriptions		Role permission: Allows a user to view event subscriptions.
View Export		Role permission: Allows a user to view the export log of the system to see when users exported secrets. Does not allow a user to export.
View Folders		Role permission: Allows a user to view, but not edit, folders in the system.
View Group Roles		Role permission: Allows a user to see which groups and users are assigned to which roles. Does not allow a user to change these assignments.
View Groups		Role permission: Allows a user to see which groups exist in the system. Also allows a user to see which users belong to each group.
View HSM		Role permission: Allows a user to view the Hardware Security Module (HSM) configuration settings.
View IP Addresses		Role permission: Allows a user to view IP Address Ranges that have been created to restrict access. Does not allow a user to edit these ranges.

Term	Definition	Description
View Jumpbox Route		Role permission: Allows a user to view the details of all jump server routes in the Admin Jumpbox Route page but not make any changes.
View Key Management		Role permission: Allows a user to view the Key Management settings (Secret Server Cloud only).
View Launcher Password		Role permission: Allows a user to unmask the password on the view screen of secrets with a launcher. Typically, this includes Web Passwords, Active Directory accounts, Local Windows accounts, and Linux accounts.
View Licenses		Role permission: Allows a user to view, but not edit, the licenses in the system.
View Lists		Role permission: View lists and list contents in Admin > Lists.
View Nodes		Role permission: Allows a user to view, but not edit, the Secret Server web server nodes.
View OpenID Connect		Role permission: View OpenID Connect integration settings in the Configuration Login tab. This replaces the Delinea One equivalent.
View Password Requirements		Role permission: Allows a user to view character sets and password requirements.
View Pipelines		Role permission: Allows a user to view event pipeline policies and policy activities.
View Platform Integration		Role permission: Allows a user to view the Secret Server connection to the Delinea platform.
View Remote Password Changing		Role permission: Allows a user to view, but not edit, Heartbeat and Remote Password Changing settings.
View Reports		Role permission: Allows a user to view, but not edit, reports. See "Browse Reports."
View Roles		Role permission: Allows a user to view roles in the system. Also allows a user to see which groups are assigned to which roles.

Term	Definition	Description
View Scripts		Role permission: Allows a user to view PowerShell, SQL, and SSH scripts on the Scripts Administration page.
View Search Indexer		Role permission: Allows a user to view, but not edit, search indexer settings.
View Secret		Role permission: Allows a user to only view which Secrets exist in the system.
View Secret Audit		Role permission: Allows a user to view Secret Audit.
View Secret Password and Private Key History	Role permission: Allows a user to see the history of passwords, private keys, or passphrases in both old and new UI.	
View Secret Policy		Role permission: Allows a user to view, but not edit, Secret Policies.
View Secret Templates		Role permission: Allows a user to view, but not edit, Secret Templates.
View Security Analytics		Role permission: Allows a user to view, but not edit, settings for Privilege Behavior Analytics.
View Security Hardening Report		Role permission: Allows a user to view the Security Hardening Report.
View Session Monitoring		Role permission: Allows a user to view active launcher sessions.
View Session Recording		Role permission: Allows a user to view recorded sessions within Secret Server.
View SSH Menus		Role permission: Allows a user to view existing SSH Menus, used in allow-listing commands that can be used on a SSH session.
View System Log		Role permission: Allows a user to only view the System Log, which shows general diagnostics information for Secret Server.

Term	Definition	Description
View Teams		Role permission: Users can view all teams. This is essentially a read-only Administer Teams.
View User Audit Report		Role permission: Allows a user to view, but not edit, the User Audit Report.
View Users		Role permission: Allows a user to view which users exist in the system.
View Workflows		Role permission: View (not edit) workflows used for multi-tier secret-access approvals and secret erase requests.
VIM	Vi improved text editor	Advanced text editor.
VM	Virtual Machine	Software emulation of a physical machine.
VMware ESX/ESXi		Built-in secret template.
VNC	Virtual Network Computing	A graphical desktop sharing system that is not unique to Windows.
VPC	Virtual Private Cloud	Isolated cloud resources.
VPN	Virtual Private Network	Secure network over the internet.
VRM	Virtual Resource Manager	Resource manager for virtual environments.
VTY	Virtual Teletype	Terminal emulation.
W3C	World Wide Web Consortium	Organization for web standards.
WAN	Wide Area Network	Network covering a large area.
WatchGuard		WatchGuard is a technology company that specializes in network security products and services, including firewalls, secure Wi-Fi, multi-factor authentication, and network intelligence solutions for small to medium-sized businesses and organizations.
WatchGuard		Built-in secret template.

Term	Definition	Description
WCF	Windows Communication Foundation	Framework for building connected systems.
Web Launcher		Web launchers are a separate login method from the Web password filler and provide a convenient click to automatically log on simpler websites. Web launchers do not work on complex login pages that rely on JavaScript. For those login pages, use the browser extension for the Web password filler. By default, Web launchers are enabled on the Web Password Secret template, but they can be enabled on custom templates as well, as described in Enabling Launchers.
Web Password		Built-in secret template.
Web Password Filler (WPF)		Web Password Filler provides easy password autofill and lifecycle management services for web applications and web sites. It allows browsers to find and enter credentials of users, when a Delinea Platform or Secret Server instance has secrets related to that website.
Web Services Impersonate		Role permission: Allows a user to send an approval request to act as another user within their organization when accessing Secret Server programmatically. Administrators do not have this permission by default.
WEBM	WebM open media format	Media file format.
Windows Account		Built-in secret template.
WMA	Windows Media Audio	Audio file format.
WMI	Windows Management Instrumentation	Windows management tool.

Term	Definition	Description
Workflows		Access Request Workflows are improved access requests that allow users to build more complex interactions based on events within Secret Server. The first release of workflows offers access requests. Workflow templates define the series of steps and reviewers required for an access request. You can assign workflows to secrets or secret policies. The original access requests are one level or step–anyone approving approves the request–no other input is required. Workflows allow up to 15 approval steps where approval by reviews in step 1 moves the request to step 2, approval at step 2 moves it to step 3 and so forth. Denial at any step denies the request.
WPF	Windows Presentation Foundation	UI framework for Windows.
WPF	Web Password Filler	See Web Password Filler.
WS	Web Service	Service available over the web.
WSDL	Web Service Description Language	XML-based language for describing Web services.
WSUS	Windows Server Update Services	Windows update management.
XML	Extensible Markup Language	Markup language for encoding documents.
XPM	EXtended Privilege Management	Management of extended privileges.
XSS	Cross-Site Scripting	Web security vulnerability.
XXE	XML External Entity	XML parsing vulnerability.
YAML	YAML Ain't Markup Language data format	Data serialization format.

Term	Definition	Description
z/OS	Zero Downtime Operating System	z/OS is a 64-bit operating system for IBM mainframes, designed for robust performance, advanced security, and superior scalability, primarily used in enterprise computing environments requiring high levels of processing power and reliability.
z/OS Mainframe		Built-in secret template.

Self-Help Resources

- Forums. Forums are oriented toward admins and other technical users.
- Delinea Blog
- "Secret Server Glossary" on page 4

Technical Support

To have access to Delinea Technical Support, you must have an equal number of unexpired user and support licenses. All support licenses expire 365 days after they are issued.

Technical Support Coverage

Please see our *Getting Technical Support* section below.

Please see the **Support** link on the menu above on delinea.com for details about our support policies.

Accessing Upgrades

Supported customers have access to all new releases (both minor and major). See .

Requesting New Features and Providing Feedback

We encourage customers with active support licensing to provide feedback on our Delinea Ideas Portal.

Getting Technical Support

Important: Please see the **Support** link on the menu above for details about our support policies.

Step One: Gather Information You May Need

Before you contact Support, gather the following information:

- Your Delinea Support username and password
- The email account already associated with your account (if using email)
- Your company name

- The technical contact name
- The technical contact phone number
- The product name
- Issue symptoms and details
- Any other relevant details, such as hours the technical contact is present

Step Two: Get a Mandatory Support PIN

Secret Server

The support PIN validates that your license includes support, and you must provide the PIN in your email or when you call. The PIN also makes it easier for Delinea Support to locate your customer records and give you better support.

To get your PIN:

1. Get the log on the credentials you received when you became a Delinea customer.
2. Log on the Support Portal using your credentials.
3. On the main page, click the large blue **PIN** bar to get your PIN. The PIN appears on the button.
4. Record your PIN.
5. If you want to use our ticketing system for support, leave the browser tab open, and return for step four.

Secret Server Cloud

The support PIN validates that your license includes support, and you must provide the PIN in your email or when you call. The PIN also makes it easier for Delinea Support to locate your customer records and give you better support. In addition, there is an additional "privileged PIN" for accessing your cloud instance.

To get your PIN:

1. Log on the Cloud Manager Dashboard.
2. Click the **Generate Tech Support** PIN button. A Tech Support PIN popup appears.
3. Record your PIN.
4. Click the Generate Privilege PIN button. Another Tech Support PIN popup appears.
5. Record your privileged PIN. Note that privileged PINs begin with "p"

   Important: Providing us a privileged PIN gives Delinea Technical Support write access to your cloud database for one day. Secrets and other sensitive data remain encrypted and unreadable.

Step Three: Choose a Support Method

Delinea customers have access to support by phone, email, and our support ticketing system (best for issue tracking). In all cases, **you must first obtain a support PIN**.

> For Severity 1 issues you **must** use phone support. Otherwise, use the method you prefer. Severity 1 means a critical problem that has caused *complete loss of service* and work cannot reasonably continue at your worksite.

Step Four: Contact Support

Click the **Support** link on the menu above for Support email addresses and phone numbers.

Using one of the below methods, contact Delinea Support.

Phone Support

Delinea delivers support by phone worldwide. Select the applicable number from the list on the **Support** link in the menu above.

Email Support

Send your email to support@delinea.com **with the PIN number as part of the subject line** of your email. For example: PIN 345 Workflow Stopped Unexpectedly. Include all the information listed in step one.

> You must send your email using an email address already noted in your account with Delinea. Otherwise, it might delay our response

Ticketing System Support

Open a support ticket and track your issue to resolution.

- Visit the Support Portal Login Page and login using the credentials you received when you became a customer.
- After logging on, click the **Log a Case** button to create a new case or click **Case Management > Cases** to manage an existing one.

Guides and Tutorials

> We also offer a range of training videos on the Delinea training site. This is a subscription service. Please contact your account manager for details.

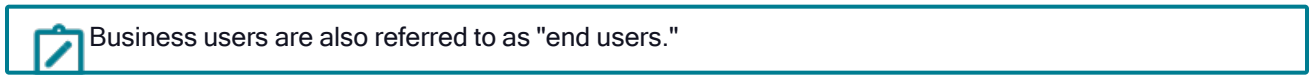
This section provides tailored information for specific groups. These include:

- Secret Server Business User Guide: This guide is for regular, non-administrative, users of Secret Server. It is mostly a set of links to a subset of the greater corpus of Secret Server documentation.
- Getting Started Tutorial Overview: Secret Server is a powerful application with many facets. As such, approaching it for the first time can be daunting. To counter that, we created this section, which is an introductory guided tutorial, for new technical users. The tutorial suggest an order to learn topics and points to specific sections of documentation for details.

- Secret Server Cloud Quick Start: Secret Server Cloud is a scalable, multi-tenant cloud platform that provides the same features as the on-premise Secret Server Professional edition. With the Secret Server Cloud, all backend services, databases, and redundancy are securely managed by Delinea and hosted on the Microsoft Azure platform. Customers do not have direct access to the databases or application file system.
- Product Overview: This is the landing page for Secret Server. It includes these topics:
  - Getting Started
  - Best Practices
  - Download Secret Server
  - Release Notes
  - Delinea Blog
  - Video Tutorials

Secret Server Business User Guide

This guide is for regular, non-administrative, users of Secret Server. It is mostly a set of links to a subset of the greater corpus of Secret Server documentation. For Secret Server Cloud, see the "Secret Server Cloud Quick Start" on page 1212.



What Is Secret Server?

Secret Server is a comprehensive Privileged Access Management (PAM) solution designed to protect, control, and manage privileged accounts and credentials within an organization. It offers a secure, centralized vault to store sensitive information, such as passwords, keys, and certificates, while ensuring that access to these critical assets is granted only to authorized personnel.

Equipped with advanced features like access control, auditing, and automated password rotation, Secret Server enables organizations to maintain a strong security posture, reduce the risk of data breaches, and comply with regulatory requirements.

What Is the Purpose of the Business User Guide?

Secret Server is a powerful, advanced product with a wide range of capabilities. Even so, it is very easy to use for regular day-to-day operations for non-technical people. The key to this is knowing what to ignore and understanding the bits you do need to know. This guide is designed to help you do just that. It provides links to only what you need to know. You can add other topics later as needed.

Getting Help

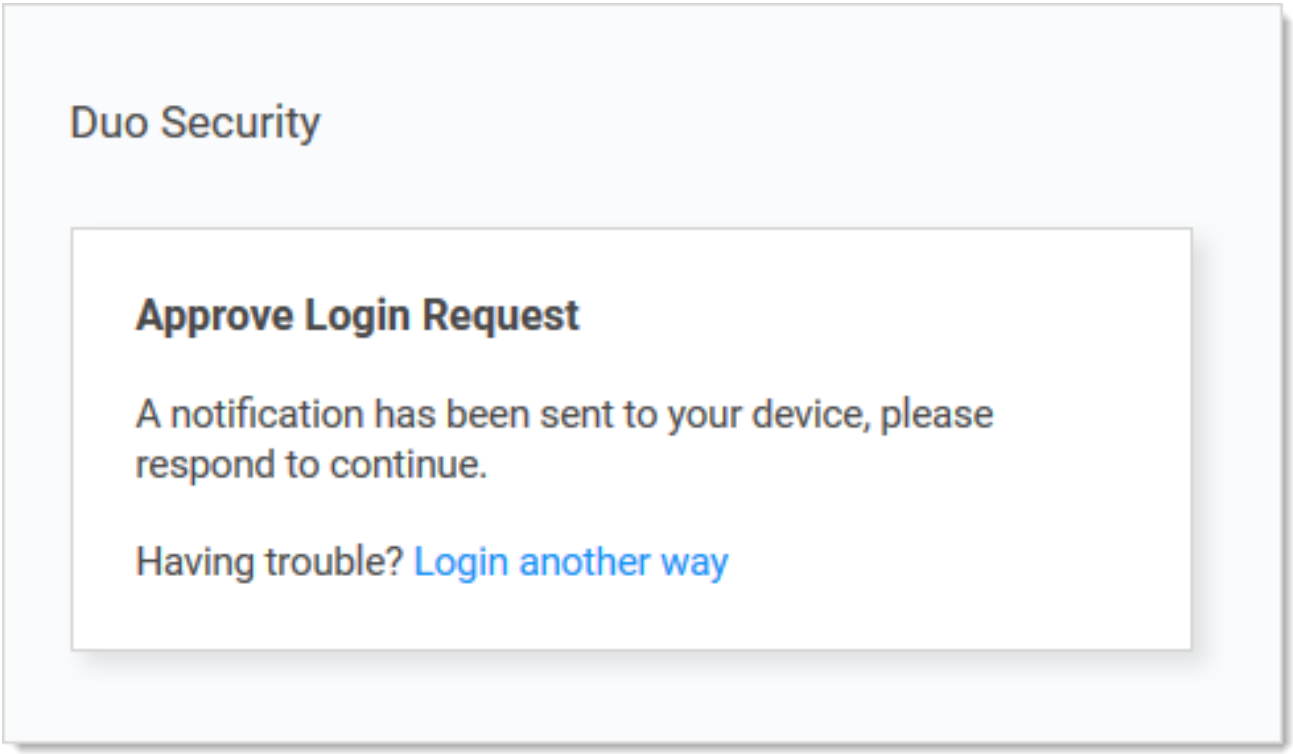
- Technical Support: Please contact your organization's help desk.
- "Self-Help Resources" on page 72
- "Document Conventions" on page 3

Important: When using this User Guide, it is easy to get lost in the ocean of Secret Server documentation. To avoid that, we recommend using **<Ctrl >** + click to access the links here. That way, the page you are going to will open to a new browser tab, leaving this one as is, making it much easier to get back to. You can also simply use the browser back button to return, but that can get tiresome because many pages link to others.

Logging on Secret Server

Depending on how your administrators configured Secret Server, you can log on with either your Active Directory account or a local account.

1. In your browser, go to the URL for your organization's Secret Server.
2. On the **Pick Your Account** popup, select your Active Directory account. The Enter Password popup appears. If you do not have an AD account, you may need to enter your local or domain information.
3. Click the **Sign In** button. If you have Duo two-factor authentication, this appears:



Your cell phone receives a notification you have to approve to access Secret Server.

> Secret Server also supports other two-factor authentication methods (depending on what your organization configured), such as text or email codes that Secret Server prompts you for.

> After you log on with your local account for the first time, you are immediately prompted to change your password .

4. Click the **Login** button. The Secret Server All Secrets page appears.

Secrets

Secrets are individually named packets of sensitive information, such as passwords. Secrets address a broad spectrum of secure data, each type represented and created by a *secret template* that defines the parameters of all secrets based on it. Secrets are very powerful and provide many ways of controlling and protecting their data, such as:

- Ensuring passwords are long, complex, and frequently changed.
- Relieving users of having to remember numerous complex passwords or when to change them. You only need to remember your password to access Secret Server. All of your secret passwords are managed for you.
- Automatically changing passwords at set intervals with no user intervention.
- Defining who has access to the secret.
- Ensuring the person accessing Secret Server or a secret is indeed you.
- Recording who actually accessed a secret.

All secret text-entry field information is securely encrypted before being stored in the database, including a detailed audit trail for access and history.

Some important basic information about secrets:

- "Viewing Secrets" on page 1147 (includes checking expiration and history)
- "Creating Secrets" on page 1127
- "Secret Configuration Options" on page 1148
- "Editing Secrets" on page 1134 (includes manually changing passwords, instead of waiting for expiration)
- "Deactivating and Reactivating Secrets" on page 1132

Secret Folders

Secret folders allow you to create containers of secrets based on your needs. They help organize your customers, computers, regions, and branch offices, to name a few. Folders can be nested within other folders to create sub-categories for each set of classifications. Secrets can be assigned to these folders and sub-folders. Folders allow you to customize permissions at the folder level, and all secrets within can inherit the folder's permissions. Setting permissions at the folder level ensures future secrets placed in that folder have the same permissions, simplifying management across users and groups.

- "Creating Folders" on page 1061
- "Moving Secrets Between Folders" on page 1068

Using Secrets on Websites (Web Password Filler)

Please set up Web Password Filler (WPF) in the following order:

1. Ensure you can log in to Secret Server the conventional way.
2. If necessary, create a folder in Secret Server where the WPF secrets will reside.
3. Install the WPF browser extension.

4. Configure WPF to point to Secret Server.
5. Login to Secret Server via WPF.

Checking out Secrets

The Secret Server *check-out* feature grants exclusive access to a single user. If a secret is configured for check out, a user can then access it. No other user can access a secret while it is checked out, except unlimited administrators. This guarantees that if the remote machine is accessed using the secret, the user who had it checked out was the only one with proper credentials at that time. See "Checkout Overview" on page 1077 for details.

Getting Notified of Secret Events

Secret Server records specific events, including expired secrets, and optionally sends you alerts when they happen. See the "Notification Inbox Overview" on page 319 and "Creating Event Subscriptions" on page 311 for details.

Learning More About Secret Server—the Getting Started Tutorial

We created a "Getting Started Tutorial Overview" below for technical users. While it covers many things you do *not* need to know right now, you may later find it helpful if you want to get a deeper understanding of Secret Server.

Getting Started Tutorial Overview

Secret Server is a powerful application with many facets. As such, approaching it for the first time can be daunting. To counter that, we created this section, which is an introductory guided tutorial, for new users. The tutorial suggest an order to learn topics and points to specific sections of documentation for details.

> This tutorial is for system administrators and other technical professionals. We recommend that non-technical users start with our "Secret Server Business User Guide" on page 75. For Secret Server Cloud, see the "Secret Server Cloud Quick Start" on page 1212.

Step 1: Trial Installation Prerequisites

> This topic only applies to **Secret Server On-Premises**.

Below are our suggested guidelines for preparing to run a trial or proof-of-concept (POC) of Secret Server.

System Requirements

Please review the detailed "System Requirements for Secret Server" on page 86. The *Minimum Requirements* are for trial, sandbox, and POC environments. The *Recommended Requirements* are for production deployments.

Hardware Requirements

Secret Server can be installed on a physical server or virtual machine.

If you would like to set up front-end (application) clustering, you need to have two or more servers available.

For testing of high availability for the SQL Server, you can use either existing Microsoft AlwaysOn infrastructure or database mirroring. If you choose to test this, this is something your database team needs to prepare in advance.

Software Requirements

Checklist

- Windows Server 2012 or newer (recommended) (one server, minimum)
- SQL Server (one instance, minimum)
- Application server prerequisites
- SSL certificate

SQL Server

> Delinea does not support using SQL Express in a production environment due to size and performance limitations.

You can create the SQL database in an existing SQL instance, or a new installation of SQL Server. For high availability, this needs to be a paid edition of SQL Server (not SQL Express). If you are using a new installation of SQL Server, please have this installed beforehand.

Detailed instructions for installation and configuration of SQL Server are included in one of the installation guides below (choose the guide matching the OS that SQL server will be installed on).

Application Server

We recommend installing Secret Server on Windows Server 2012 or greater. Include IIS, ASP.NET and .NET Framework. Refer to the System Requirements above to view prerequisite details.

Application Configuration

Service Account

Set up a service account:

1. Log on as a batch job (on the server that Secret Server runs on)
2. Modify permissions to the Secret Server application directory (typically `C:\inetpub\wwwroot`) and `C:\Windows\temp`.
3. Provide access to your SQL Server instance by adding the db_owner permission to the Secret Server database.

For detailed instructions on how to configure the permissions for the service account, see "Running the IIS Application Pool As a Service Account" on page 60. The installation guides include instructions for assigning db_owner permission to the service account in SQL Server.

If you would like to test features that rely on Active Directory, such as AD group sync or discovery, you should also have accounts available with the appropriate permissions (described below). One option is to use the same account for both features.

Active Directory Group Sync

Active Directory group synchronization means that Secret Server can automatically add users and enable or disable them to log into Secret Server based off of their Active Directory group membership. You can choose which groups to sync. When configuring AD group sync in Secret Server, you are required to specify an account that can read the properties of users and groups. See "Active Directory Rights for Synchronization Account" on page 493 for a detailed list of required permissions.

Discovery

To test discovery, please have some machines available for Secret Server to connect to for discovering accounts. An account is required to sync with AD and also scan the machines found for Windows local account and service account discovery. "Account Permissions for Discovery" on page 528 describes the permissions required for an AD account to be used for discovery.

Test Accounts

We recommend having a few test accounts available to represent the types of accounts you want to manage using Secret Server. These could be local Windows accounts, service accounts running scheduled tasks or services, SQL server accounts, and others.

Email Notifications

To test email notifications, which can be used for event subscription notifications or requests for approval to passwords, you need configuration information for the company SMTP server:

- Service account to run the application and connect to SQL
- Domain (test or production)
- Domain account to be used for AD sync and discovery
- Test machines (if testing discovery)
- Test accounts
- SMTP server settings

SSL Certificate

We recommend setting up SSL (or https) for access to Secret Server. To do so, you will need an SSL certificate. You may use an existing wildcard certificate, create your own domain certificate, or purchase a third-party SSL certificate for Secret Server.

Firewalls and Ports

Secret Server must connect directly to a target system to change its password. For devices that are firewalled off from Secret Server, remote agent can provide connectivity to them, but they also require connectivity from them to the target systems for password changing.

Please see "Ports and IP Addresses Used by Secret Server" on page 765 for a list of ports needed by Secret Server for password changing, discovery, and other features.

Step 2: Installation

This topic only applies to **Secret Server On-Premises**.

Process

Run the Installer: Secret Server comes with an installer that walks you through the entire process from start to finish. Once you have the prerequisites ready to go, download and run your installer, and the wizard will take you through the installation process. Please see our "Installation" on page 65.

Licenses

See the "Licensing" on page 80 section.

Step 3: Secret Server Dashboard

The Secret Server Dashboard is the main page for searching and viewing secrets. Nearly everything you do in Secret Server starts with the Dashboard. See "Application Dashboard" on page 157 for details.

Step 4: Security Best Practices

As you start using Secret Server, we strongly recommend configuring the following security settings. While these re optionl, setting them is a a best practice.

Local Admin Account Best Practices

Even if you plan to "AD and Secret Server Overview" on page 491 to log into Secret Server, chances are you will need to use this account again. This is the first account you created during the installation process. Keep this account secure and avoid being locked out of Secret Server by following these suggestions:

- Store the credentials in a secure location that you can access if you lose all access to Secret Server.
- Enable the **Allow Users to Reset Forgotten Passwords** setting to provide a way of resetting the password if account is locked out or if the password is forgotten:
  1. Select **Admin > Configuration**. The Configuration page appears.
  2. Click the **Local User Passwords** tab to locate the setting.
  3. Click the **Edit** button to edit the setting.
  4. Click the **Save** button when finished.

This requires having an SMTP server configured.

- Configure the other **Local User Passwords** settings to enforce your password requirements, expiration, password history, and other password policies.

SSL (HTTPS) Best Practice

We recommend requiring SSL access to Secret Server. This requires setting up an SSL certificate for the website, preferably with a domain certificate. However, if you don't have a certificate, see "Installing Self-Signed SSL Certificates" on page 430. Once you have your certificate:

1. Configure the HTTPS binding for your Secret Server website using the certificate you choose.
2. Ensure your certificate is trusted on the Secret Server users' machines. See "Trusting an SSL Certificate on a Client Machine" on page 432 for instructions.
3. Enable **Force HTTPS/SSL** on the **Security** tab of the Secret Server**Configuration** settings.

Step 5: Backups

Configure backups to avoid losing your data. Secret Server provides the option to automatically take a backup on the interval you specify, sending the backups to a local or network location. There are two components of an entire backup of Secret Server: the Web application files and the database. Find these settings by selecting **Backup** from the **Admin** menu. See "Backup and Disaster Recovery" on page 451 for more information.

To configure the backup paths, see "Backup Settings" on page 453 .

The file paths configured on this page by default need to be either changed or created on each server that the Secret Server application and database reside on.

Step 6: Active Directory Integration

This topic only applies to **Secret Server On-Premises**.

To allow users to log in with their Active Directory (AD) credentials, you can configure your AD domain settings in Secret Server and then add users either individually or by group.

Setting up Active Directory

See "Configuring Active Directory" on page 503.

Enabling Active Directory Users

See "Enabling and Disabling Active Directory Users" on page 506.

Managing Active Directory Users via a Distributed Engine

See "Syncing and Authenticating AD Users via a Distributed Engine" on page 508.

Step 7: Secret Server Framework

To try out Secret Server, you must have folders, roles, users, and secrets to operate on:

1. Setup some folders and roles: We encourage is for you to setup a folder structure and a few roles. The folder structure is how you will keep your secrets organized, and provide access to shared secrets. Additionally, roles ensure you are able to control access to different parts of Secret Server and assign permissions to view certain folders and secrets. See "Folders" on page 1059 and "Overview of Users, Roles, User Groups, and User Teams" on page 1271.
2. Add users if you have not already from AD. See "Creating Users" on page 1273 and "Creating User Groups" on page 1284.

3. Add an Active Directory or other secrets. If you plan on using discovery, the account will also need permissions to scan computers on the network for accounts. See "Secret Management Overview" on page 1117.

Step 8: Discovery

Secret Server has a discovery feature that can automatically find local Windows accounts, Active Directory service, Unix, VMware ESX/ESXi, and Active Directory domain accounts. Account and dependency types not supported out-of-the-box in Secret Server can still be discovered by writing PowerShell scripts that can be run as custom scanners. This allows administrators to quickly import accounts found by Secret Server on specified domains or IP addresses.

Please see the "Discovery Overview" on page 522 for a comprehensive guide to configuring and using discovery.

To run discovery on a domain, IP address range, or a custom source, you need to first enable the discovery feature for Secret Server. Second, you must enable discovery for each discovery source you would like to be scanned.

Step 9: Remote Password Changing

Secret Server remote password changing (RPC) provides the ability to either start a password change manually or schedule automatic password changes to occur at a regular interval.

Enabling Remote Password Changing

See "Enabling RPC" on page 921.

Performing a Manual RPC

See "Running a Manual RPC" on page 921.

Common RPC Error Codes

See "RPC Error Codes" on page 948.

Step 10: Heartbeats

Heartbeat allows you to determine from Secret Server whether the credentials in a secret authenticate successfully with their target system. By default, heartbeat is turned off in Secret Server. See "Heartbeat Overview" on page 1042 for general information.

Enabling Heartbeat

See "Enabling Heartbeat in RPC" on page 1044.

Running Heartbeat

See "Running Heartbeat for a Secret" on page 1046.

Step 11: Audits and Reports

Before running reports and audits, you must create something to report on—to that end:

- Import a few accounts or create secrets manually
- Rotate passwords a few times
- View a couple of your secrets

This generates enough audit logs to provide meaningful outputs in your reports:

- Security Hardening Report
- What secrets have been accessed
- What secrets failed heartbeat
- Failed login attempts
- Secret activity

See "Built-in Reports" on page 883 for the most up-to-date list of reports included.

For details on using reports, see:

- "Creating and Editing Reports" on page 887
- "Viewing Reports" on page 903

Step 12: Secret Access and Workflow

Sometimes, depending on your scenario, you want to add extra protections to highly sensitive secrets. Secret Server has a access request and workflow features:

- "Checkout Overview" on page 1077: Grant access to a single user
- "Access Request Overview" on page 1074: Require approval prior to accessing a secret for a defined time period
- "Workflow Overview" on page 1089: Require multi-level and multi-user approval prior to accessing a secret for a defined time period
- "QuantumLock Overview" on page 1080: Add another security layer by encrypting secret data with a supplemental custom encryption key that is only accessible with an additional password, regardless of regular permissions.

Step 13: Secret Launchers

A secret *launcher* opens a connection to the remote computer or device or logs into a website using the secret's credentials directly from the Web page. While this provides a convenient method of opening RDP and PuTTY connections, it also circumvents users being required to know their passwords. A user can still gain access to a needed machine, but it is not required to view or copy the password out of Secret Server. A Web launcher automatically logs into websites using the client's browser. SS launchers, also called protocol handlers, come in three primary types:

- **Remote Desktop:** Launches a Windows Remote Desktop session and automatically authenticates the user to the machine.
- **PuTTY:** Opens a PuTTY session and authenticates the user to a Unix system.

- **Web Password Filler:** Uses a Chrome extension to automatically log the user into a website with secret credentials. See our separate documentation for Web Password Filler.
- **Web Launcher:** An alternative method to automatically log on websites. See "Web Launchers" on page 708.

See "Overview of Secret Launchers and Protocol Handlers" on page 658 for more information.

Step 14: Recording Sessions

Session recording provides an additional level of security by recording a user's actions after a launcher is used. Session recording works for any launcher, including PuTTY and SSH, Windows Remote Desktop, Microsoft SQL Management Studio, and custom executables. The resulting movie is viewable from the secret audit. There are two types of session recording:

- "Session Recording Overview" on page 1226
- "Session Recording Overview" on page 1226

Step 15: Secret Server APIs and CLI

You can access Secret Server without using the user interface for automation and integration purposes. Currently, there are two APIs:

- An asynchronous REST (representational state transfer) API for Web services, which is based on JSON (JavaScript Object Notation). This is the preferred method. It is faster and easier to read than the SOAP API and is still actively updated.
- A synchronous SOAP (Simple Object Access Protocol) for Web services, which is based on XML. This method is deprecated, but we still support it. It is based on an older technology, which has largely been replaced in recent years. There will be no enhancements to this API. There are, however, a few, rarely used capabilities that only our SOAP API has.

We offer a software development kit (SDK) that contains a .NET framework and a command line interface (CLI) for accessing the REST API with Windows applications or scripting languages.

Both APIs, the .NET framework, and the CLI support:

- GET Requests: Retrieve information from Secret Server, including entire secrets, individual secret fields, and security tokens
- POST Requests: Create Secret Server data
- PUT Requests: Update Secret Server data
- DELETE Requests: Remove Secret Server data
- Once-per-session permissions (tested once and then based on the IP address), administered with a Secret Server rule

SDK Documentation:

- "Using the Secret Server SDK for DevOps" on page 1533: Includes these topics:
  - Secret Server configuration
  - Roles and permissions
  - SDK client installation
  - Connecting to Secret Server
  - SDK client caching
  - Examples
- "Downloads for the Secret Server SDK for DevOps" on page 1551: Includes these topics:
  - SDK downloads
  - Download
  - SDK release notes
  - NuGet packages
- SDK Integration Document: Includes these topics:
  - Integrating using C#
  - Integrating using the `web.config` file
  - Methods of the SecretServerClient() class

REST API Documentation:

- "REST API Reference Download" on page 1500: Links to online reference guides (by Secret Server release)
- "REST API PowerShell Scripts" on page 1503
- REST API Python Examples

SOAP API Documentation:

- "APIs and Scripting" on page 1466: Reference guide in a downloadable PDF

Step 16: Additional Resources for Secret Server

You have finished this "Getting Started" introduction to Secret Server. There is much more to explore within Secret Server, such as scripting, third-party Integrations (SIEM, CRM, HSM, and more), and connecting to Privilege Manager to monitor and protect endpoints. We look forward to working with you!

See "Help" on page 2 to learn more about Secret Server and other Delinea products.

Business Users vs IT Users

What is a Business User?

A Business User refers to someone who is not an IT user. These users generally utilize Secret Server to securely store credentials needed for accessing departmental websites, applications, and personal accounts within a highly secure enterprise vault. Business Users can:

Business users can:

- Access secrets: They can create, update, and delete their own secrets within Secret Server. For example, a user signing up for an online service can use our password generator to create a strong password, store that password in Secret Server, and later use Web Password Filler to access it when logging on.
- Request and approve access to secrets: Non-privileged secrets may need approval workflows. Business users can request access to these secrets and can approve access for others. For example, if access to an organization's social media accounts required authorization from a member of the marketing team, a business user could request access to the secret, and another business user could then approve access.
- Share secrets with other users: Business users can share non-privileged secrets with other users of Secret Server, either IT or business users.
- Access secrets using our mobile app: Business users can use the Delinea mobile application to access and manage their secrets.
- Use Web Password Filler: Business users can use our Web Password Filter (WPF) for accessing websites and auto-filling credentials from their secrets.
- View audits of their secrets: Business users can use auditing to see who accessed their shared secrets.

Administrators can restrict template visibility for Business Users. By default, all users have access to all templates, but business users typically need access to the following templates:

- Bank Account
- Combination Lock
- Contact
- Credit Card
- Healthcare
- Password
- PIN
- Security Alarm Code
- Social Security Number
- Web Password (Non-IT-related)

What is an IT User?

An IT User is an individual with a user account within the system who is part of the IT team and has access to manage sensitive information like passwords, API keys, and other credentials stored within Secret Server, allowing them to securely access and manage privileged accounts for various systems across the organization, typically with specific permissions based on their role within IT.

The table below shows available actions to Business Users and IT Users:

Action	Business User	IT User
Approve access to secrets	•	•

Action	Business User	IT User
Configure password rotation		•
Configure security features for a secret		•
Create, delete, modify folders and add secrets to any folder		•
Create and delete personal folders	•	•
Create, update, and delete personal secrets	•	•
Create and manage Integrations, workflows, pipelines, discovery, sites, distributed engines, HA/DR, and more		•
Launch secrets they have access to	•	•
Request access to secrets	•	•
Share secrets	•	•
Use any administrative functions of Secret Server		•
Use any launcher including RAS (Platform)		•
Use Connection Manager to launch RDP or SSH sessions		•
Use the Delinea mobile app for access to secrets	•	•
Use the Secret Server SDK or API		•
Use Web Password Filler	For Non-IT-related tasks, websites or portals	For any tasks
Use Web templates in Secret Server	For Non-IT-related tasks, websites or portals	For any tasks
Utilize any templates including custom templates		•
View secret audits for their own secrets	•	•
View user audits for their own secrets	•	•

Secret Server Documentation

Introduction

Delinea Secret Server is an enterprise-grade password management solution designed to help organizations securely store, manage, and control access to privileged credentials. It aims to improve the security of sensitive data, reduce the risk of data breaches, and streamline the password management process.

Here are the key features of Delinea Secret Server:

- Secure Password Storage: Secret Server stores privileged credentials in an encrypted format, protecting sensitive information from unauthorized access.
- Access Control: Secret Server implements role-based access control, allowing administrators to set permissions and control who has access to sensitive information.
- Privilege Escalation Management: Secret Server integrates with Windows systems to provide privilege escalation management, helping to reduce the risk of data breaches.
- Auditing and Reporting: Secret Server provides detailed audit logs and reports, making it easier for organizations to track access to sensitive information and detect any unauthorized activity.
- Automated Password Management: Secret Server supports automated password management, helping to streamline the password management process and reduce the risk of manual errors.
- Multi-Factor Authentication: Secret Server supports multi-factor authentication, helping to improve the security of sensitive information.
- Integration with Other Tools: Secret Server integrates with a variety of other tools, including Active Directory, Microsoft Azure, and cloud-based applications, making it easier for organizations to manage their passwords and access controls.

This section of the Delinea documentation portal supports Secret Server.

> Navigate using the dynamic table of contents on the left, the page contents on the right, or by entering a search term above. Many pages in this documentation have sub-pages. The container (parent) pages can have introductory text or simply a heading with no text. Please click the table of contents on the left to see any sub-pages it might have.

Documentation

You are at the home page of the current Delinea Document Portal for Secret Server. It contains:

- New material.
- Links to legacy knowledge bases article that have yet to be converted or archived. There are very few of these left.
- Links to legacy PDF documentation. These are also rare, and if you do find one, its target is very likely out of date.

Getting Started

- "Secret Server Business User Guide" on page 75 (for non-technical users)
- Getting Started Tutorial (for technical users)
- "Installation" on page 65
- "System Requirements for Secret Server" on page 86

Best Practices

- "Configuration Best Practices" on page 14
- "Discovery Best Practices" on page 550
- "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310 (PDF)
- "Security Hardening Guide" on page 1402

Download Secret Server

Product Downloads

Release Notes

"Secret Server Release Notes" on page 1566 (On-Premises and Cloud)

Delinea Blog

The Delinea blog offers cybersecurity insights and expertise from industry leaders. With a focus on zero trust, identity management, and privileged access, the blog provides practical advice and thought leadership to help organizations strengthen their security posture. As a trusted voice in cybersecurity, Delinea leverages the blog to empower customers embracing zero trust and demonstrate their commitment to innovation.

Video Tutorials

We offer a range of training videos on the Delinea training site. This is a subscription service. Please contact your account manager for details.

Developer Resources

"Developer Resources" on page 1473

Secret Server Cloud Quick Start

Overview

Secret Server is a scalable, multi-tenant cloud platform that provides the same features as the on-premise Secret Server Professional edition. With Secret Server Cloud, all backend services, databases, and redundancy are securely managed by Delinea and hosted on the Microsoft Azure platform. Customers do not have direct access to the databases or application file system.

End users are also referred to as "business users."

Cloud Versus On-Premise Secret Server

For documentation purposes, Secret Server Cloud is the same as the corresponding on-premise edition. However, there are some feature differences:

- **Site Connectors:** On-premise versions can use multiple site connectors to manage engine connections, such as RabbitMQ or MemoryMQ. The cloud version manages this for you as an Azure service and is not configurable.
- **CRM Integration:** On-premise versions can integrate with CRMs via direct database connections or the ConnectWise API. This is not currently available in Secret Server Cloud.

Getting Started

This section walks you through an initial configuration of your cloud instance. To see additional documentation for Secret Server Cloud features, please refer to the support resources section at the end of this document.

System Requirements

All cores are physical unless otherwise noted.

A distributed Engine server is required to communicate with Secret Server Cloud. Distributed engine server recommended specifications:

A distributed engine server is required to communicate with SSC. Distributed engine server recommended specifications:

- Windows Server 2016 or Above
- CPU: 4-core 2 GHz (minimum)
- Memory: 4 GB of RAM (minimum)

Engine Connectivity

Secret Server Cloud's Architecture Diagram shows the network topology of your cloud instance. Your on-premises distributed engines do not need any inbound TCP/IP ports open (unless using RADIUS authentication). If you do not have outbound firewall policies in place, no firewall configuration is necessary. If you do, the distributed engines need outbound access to:

- Secret Server Cloud's multi-tenant front-end Web server
- A shared service bus
- A customer-specific service bus
- A Content Delivery Network (CDN)

The protocols and endpoint details are in the architecture diagram mentioned above.

Initial Setup

After you sign up for a trial, you can choose your URL name and provision your instance:

> To see additional documentation for Secret Server Cloud features, please refer to the support resources section at the end of this document.

1. After you sign signed up for a Secret Server Cloud trial, you received an email from Delinea Sales. Click the **Cloud Portal** link in that email to begin your setup. The Setup Page appears in your browser.
2. Choose your location in the **Cloud Environment** dropdown list.
3. Click the **Continue** button. The Delinea One Portal appears.
4. Create the password for your first user account with administrator credentials. This account will be assigned to the email address you entered to request the trial.
5. After confirming the password, click the **Set Password and Login** button. The Delinea log on page appears.

   > This is the backup admin account that you may need in a "break the glass" or unlimited admin situation. Delinea recommends you store the password in a secured physical location such as a safe or locked file cabinet. You can reset the password using an email reset, but **if this password is forgotten or you no longer have access to the email account, Delinea cannot reset this password**.

6. Click the blue button that matches the location you just chose. A setup page appears.
7. Type a name for your subdomain. Do not use special characters or spaces.
8. Read the Business User (End User) License Agreement.
9. Click to select the check box to signify agreement.
10. From the dropdown, select **Yes** or **No** to signify your organization's oversight of EU information.
11. Click the **Accept** button. It may take several minutes for your new Secret Server Cloud to spin up.
12. When initialization is complete, click go to your Secret Server Cloud URL and click the **Login with Delinea One** button. You are automatically redirected to your new Secret Server Cloud dashboard.

> For information on how to install a distributed engine, please refer to "Distributed Engine Installation " on page 763.

Configure Active Directory Integration

Active Directory integration allows users to log in with their domain credentials. Connections to your domain are routed through the distributed engine service running in your network.

1. On the dashboard, create a new Active Directory secret from the create secret widget in the upper right hand corner.

> The domain account should be able to read users and groups from the domain you want to sync. For detailed information on the rights required, please see the "Active Directory Rights for Synchronization Account" on page 493.

2. Type the domain, username, and password in the **Create Secret** form.
3. Save the secret.
4. Navigate to **Admin > Active Directory**.
5. Click **Edit** and check the boxes for **Enable Active Directory Integration** and **Enable Synchronization of Active Directory**.
6. Click the **Save** button.
7. Click the **Edit Domains** button.
8. Click the **Create New** button.
9. Type your FQDN and a friendly domain name that users will see on the login page.
10. Click **Sync Secret** to select the secret you just created.

> The domain site is set to default. This means that the Active Directory authentication and synchronization will run through the distributed engine service installed on your network.

> Do **not** select "Enable Login from AD." If you do, you cannot set the domain groups later in this instruction.

11. Click the **Save and Validate** button.
12. Click the **Back** button.
13. Click the **Edit Synchronization** button. The Synchronization Edit page appears.
14. In the **Available Groups** list, click each domain group that you want to log on in Secret Server Cloud instance and click the **<** button to move the group to the **Synchronized Groups** list.
15. Click the **Save** button.
16. Click the **Synchronize Now** button to start the user and group synchronization immediately. The synchronization process runs automatically, but to get immediate results, you can start it manually.

Test Heartbeat and Remote Password Changing

Heartbeat ensures the secrets you have stored have the correct password, and Remote Password Changing (RPC) changes passwords on demand or a schedule.

1. Navigate to **Admin > Remote Password Changing**.
2. Click the **Edit** button.
3. Click to select the **Enable Remote Password Changing** and **Enable Heartbeat** check boxes.
4. Click the **Save** button.

5. Click the **Run Now** button in the **Remote Password Changing and Heartbeat Log** sections. This runs the heartbeat and RPC processes immediately.
6. Go to the secret you created for domain synchronization in the previous section or create a new test secret to use.
7. A brand new secret's **Last Heartbeat** status should be pending or processing. Once heartbeat completes you should one of these statuses:
   - **Unable to Connect:**Secret Server could not reach the target machine. This could be a firewall issue or the machine name or IP address is wrong.
   - **Failed:**Secret Server could connect but could not authenticate. This likely means the password on the secret is incorrect.
   - **Success:**Secret Server successfully connected with the username and password.
8. You can test password changing by viewing a secret and clicking the **Change Password Remotely** button.

   
   This will change the password on the target system.
9. You can view the status of password changes and heartbeats in the log at **Admin > Remote Password Changing.**

Next Steps

- Add another user to the Administrator role in Secret Server. This allows you to have another administrator besides the initial user account created. To assign roles, go to **Admin > Roles** and click the **Assign Roles** button.
- Add a folder and share it with the group you synchronized from Active Directory. Create and edit folders from the Folder Tree View on your Dashboard.
- Create a secret in that folder for other users to see. When creating a secret, you can click the **Folder** link to save it to another folder.
- Have other users log on. Any users synchronized to Secret Server through the domain synchronization can log on with their domain credentials.
- Enable Google two-factor authentication by going to **Admin > Users**, editing the specific user, and assigning a two-factor option.

Troubleshooting and Resources

Get Error: "Site (Default) engines are not currently online" When Saving Domain

This can occur when Secret Server was not able to complete a round trip with the installed engine service. This validation may take several minutes for Secret Server to perform after the engine has been approved and assigned to the site. To address the issue:

1. On the server you installed engine on, check the logs in the install directory `C:\Program Files\Thycotic Software Ltd\Distributed Engine\log`.

2. If you see a message for "Could not configure, trying in 30 seconds" or a "Bus Broken Down Error" verify that the engine is approved and assigned to your default site.
3. Go to the site under **Admin > Distributed Engine > Manage Sites**.
4. Click the **Validate Connectivity** button.
5. If a success message appears and the engine status shows as online, try saving the domain again.

Secret Server Character Limits

Secret Server allows the following number of characters for the fields:

- Folder Name: 128. Error Message when exceeded - *The folder name must be 128 characters or less*.
- Group Name: 250. Error Message when exceeded - *Application Error*.
- Role Name: 255. Error message when exceeded - *Please choose a Role name with between 1 and 255 characters*.
- Secret Name: 1,992. Error Message when exceeded - *Invalid Secret Name*.
- Secret Text Fields: 9,999. Error message when exceeded - *Secret item value exceeds max length characters.*
- Secret Note Field: 9,999
- Request field 600 (This is a comment field used when requesting access to a secret). No error message given, it prevents exceeding 600 characters. If you paste in more than that it clips the text and still lets you save.
- Local User Username: 128
- Local User Display Name: 256
- Local User password: 500

Setup

This section includes setup-related topics, including:

- Secret Server Features
- Configuration Best Practices
- Download Hashes
- Downloading Secret Server
- IIS
- Installation
- Licensing
- Prerequisites
- RabbitMQ

- SQL Server
- Uninstalling Secret Server
- Upgrading

Secret Server On-Premises Features by Version

This topic only applies to **Secret Server On-Premises**.

See Viewing Your License section to check which license version you are currently using.

General

Feature	Vault	Professional	Platinum
Deployment	On-Premises for current customers Cloud for new customers	On-Premises	On-Premises
Secrets	No Limits	No Limits	No Limits
Users	25 Users	Licensed by User	Licensed by User

Access Control

See Gain control over web apps and cloud management platforms for more information.

Feature	Vault	Professional	Platinum
Password Hiding	•	•	•
RDP and PuTTY Support	•	•	•
Role-Based Access Control	•	•	•
Web Password Filler	•	•	•

Password hiding refers to techniques used to prevent passwords from being displayed or stored in plain text, thereby protecting them from being easily accessed or compromised.

Advanced Scripting

See Integrate custom and 3rd-party apps for more information.

Feature	Vault	Professional	Platinum
CLI Tools (Win, Linux, Mac)		•	•
Custom Ticket-System Integration		add on	•
Extensible (script-based) Discovery		add on	•
PowerShell Dependencies		add on	•
PowerShell Password Changing		add on	•
SDK		•	•
SQL Dependencies		add on	•
SSH Dependencies		add on	•
Web Services API		•	•

Advanced Unix Features

See Generate, store, rotate, limit the use of and manage SSH Keys for more information.

Feature	Vault	Professional	Platinum
Allowed command lists		add on	•
Blocked command lists		add on	•
SSH Key Authentication		add on	•
SSH Key Management		add on	•

Approval Workflow

See Meet regulatory requirements and demonstrate compliance for more information.

Feature	Vault	Professional	Platinum
Checkout (OTP)		add on	●
Native Ticket-System Integration		add on	●
QuantumLock		add on	●
Request Access		add on	●
Require Comment		●	●

Automation

See Core PAM Automation Increases Visibility, Control, and Oversight for more information.

Feature	Vault	Professional	Platinum
Automatic Password Changing for Network Accounts		●	●
Email Notifications	●	●	●
Heartbeat		●	●
If/Then Automation	●	●	●
Secret Policies		●	●

Discovery

See Find Unknown and Unmanaged Privileged Accounts for more information.

Feature	Vault	Professional	Platinum
AWS Discovery		●	●
Discovery Rules		add on	●

Feature	Vault	Professional	Platinum
Google Cloud Discovery		•	•
Local and Active Directory Privileged Account Discovery	•	•	•

Enhanced Auditing, Reporting, and Compliance

See Meet regulatory requirements and demonstrate compliance for more information.

Feature	Vault	Professional	Platinum
Auditing and Reports	•	•	•
Custom Reports		•	•
Dual Control		•	•
Event Subscriptions		•	•
FIPS Compliance		•	•
Scheduled Reports		•	•

High Availability and Disaster Recovery

See Disaster recovery capabilities for IT emergency scenarios for more information.

Feature	Vault	Professional	Platinum
Resilient Secrets (DR)		add-on	add-on
Unlimited Admin Mode for Emergencies ("break the glass")	•	•	•

Integrations

See Delinea Integrations Marketplace for more information.

Feature	Vault	Professional	Platinum
Access Rights Management		•	•
Behavior and Reputation		•	•
Device Management		•	•
DevOps		•	•
IBM z/OS Integration			•
Identity and Access Management		•	•
Multi-Factor Authentication	•	•	•
SAML Integrations		•	•
SAP Integration		•	•
SIEM Integration		•	•
Ticket System		•	•
Vulnerability Management		•	•

Cybersecurity "behavior and reputation" refers to the analysis and scoring of user behaviors and digital entities (such as websites, IP addresses, and files) based on their observed actions and characteristics to assess their potential risk or trustworthiness.

Secure Vault and Password Manager

See Protect enterprise privileged accounts with military-grade security for more information.

Feature	Vault	Professional	Platinum
Active Directory Integration	•	•	•
AES 256 Encryption	•	•	•
File Attachments	•	•	•
Granular permission control	•	•	•
Import and Export	•	•	•
IP Address Restrictions	•	•	•
Multi-Factor Authentication	•	•	•
Remote Access Service		add on	add on
Smartphones and Devices	•	•	•

Service Account Governance

See Automatically Change Passwords on a Schedule for more information.

Feature	Vault	Professional	Platinum
Service Account and Dependency Management		add on	•
Service Account Discovery		add on	•

Session Monitoring and Control

See Record & Monitor Privileged Session Access and Log Keystrokes for more information.

Feature	Vault	Professional	Platinum
Connection Manager		add on	add on

Feature	Vault	Professional	Platinum
Keystroke Logging		50-Secret Limit	•
Proxying RDP and SSH		•	•
Session Monitoring		add on	•
Session Recording		50-Secret Limit	•
Windows Desktop Apps			•
Advanced Session Recording Agent		add on	•

Configuration Best Practices

Getting Started

Overview

This document was written after helping many customers successfully deploy Secret Server in their organizations. It covers the issues that most customers tackle as they consider which data to store, who needs access, what permissions to apply, and how to organize all their sensitive data. This document is not meant to cover everything

Think of Secret Server as a platform for your organization to store all of its passwords and sensitive data. This means that it can be configured to work in many different ways depending on your industry, compliance requirements, and ultimate end goals. The trick is to know your objectives and then match the capabilities and best practices to your situation.

Terminology

Throughout this topic, certain terms are used to refer to specific features or concepts within Secret Server. Some of these terms corresponds to explicit roles defined within Secret Server that may be referenced, while others are broader terms that system administrators should be familiar with.

Administrator

Access to all the features within Secret Server can be granted to users by creating and assigning different roles. *Administrator* is one of the default roles that comes installed with Secret Server. By default, this role contains all role permissions, but it can be customized as well. In this guide, when it is used in the context of a Secret Server user, it is referring to the users who generally have most permissions and manage the system. Administrators have control over the global security and configuration settings.

> Administrators in Secret Server do **not** automatically have access to all data stored in the system—access to data is still controlled by explicit permissions on that data.

Basic User Role

The basic user role is a default role that comes installed with Secret Server. This role is a slimmed down version of the user role and primarily focuses on creating and modifying secrets, as well as limited "view" permissions. Users that have this role assigned to them also have their own personal folder.

Folder

A folder in Secret Server provides a hierarchical structure for organizing secrets. Some folders contain no secrets at all and may be used only to set permissions or policies on subfolders. Other folders may simply be a way to organize sub-folders that contain secrets. Folders are organized based on a "root" level folder structure, where "/" is the root level folder and any new folder created will be placed under that folder. Personal folders are unique and are created for each user, providing them the "personal folders" permission. Personal folders can contain sub-folders for the owner to organize their secrets.

Role Based Access Control (RBAC)

Secret Server role based access control (RBAC) is a mechanism that restricts system access to authorized users and defines what type of access a user has within the system. Often these roles correspond to features within the product and those features may give users greater privileges to make changes within the system. RBAC is a core Secret Server feature.

Secret

A secret is any sensitive piece of information (typically a password) that you would like to manage within Secret Server. Typical secrets include (but are not limited to) privileged passwords on routers, servers, applications, and devices. Files can also be stored in secrets allowing for storage of private key files, SSL certificates, license keys, network documentation, or even a Microsoft Word or Excel document.

Site

A site is a logical work container that can tell Secret Server which distributed engines should manage work associated with specific tasks. Sites are critical to ensuring that Secret Server can manage remote network segments, alternate locations, or even DMZs. By default, Secret Server comes with the "local" site. That site is unique as it is the only site that can be configured for "web processing" or "engine processing." When the local site is configured for Web processing, the Web servers themselves act as distributed engines and are responsible for all engine work processing, in addition to the Web Server role specific work that they may be configured for. Any additional sites that are user created may only be configured for Engine processing. The "Local" site comes with two free engines under any licensing model that may be used. Any additional sites and engines must be licensed separately and will incur additional licensing costs.

User

This is the default role for new users that are added to Secret Server. By default, this role contains several permissions that enable new users to interact with Secret Server. Many of these permissions are centered around

creating and modifying secrets, as well as several "view" permissions to access audit information. Additionally, access to advanced secret options, assigning secret policies, and a few other advanced permissions are assigned to this role. It also gives each user their own personal folder that is accessible only by each individual user added to the system. Besides the owner, only the "unlimited administrator" role can access these folders.

Know Your Edition

As you read through this guide, some features may be referenced that are only available in certain editions of Secret Server. To get an idea of what's available, you can reference the "Secret Server On-Premises Features by Version" on page 8.

Installation and Configuration

Installation

Before installing Secret Server, be sure to look at the "System Requirements for Secret Server" on page 86 The process for installing Secret Server is outlined in the "Installation" on page 65 matching the version of Windows Server you are using. If you have an active trial or have purchased Secret Server licenses, you can find your licenses by logging into your account through Cloud Manager.

Basic Configuration

Once Secret Server is installed, see the "Secret Server Business User Guide" on page 75 to begin setting up Secret Server right away. This covers:

- Adding your licenses
- Basic security settings
- Configuring automatic backups
- Basic security settings
- Heartbeat
- Basic security settings
- Setting up access for local and AD users

Advanced Configuration

Secret Server's Advanced Configuration page is intentionally hidden from casual access. You have to enter a URL—the page is not accessible by clicking a link. The URL format is:

```
https://<>/app/#/admin/advanced-config-settings
```

For example:

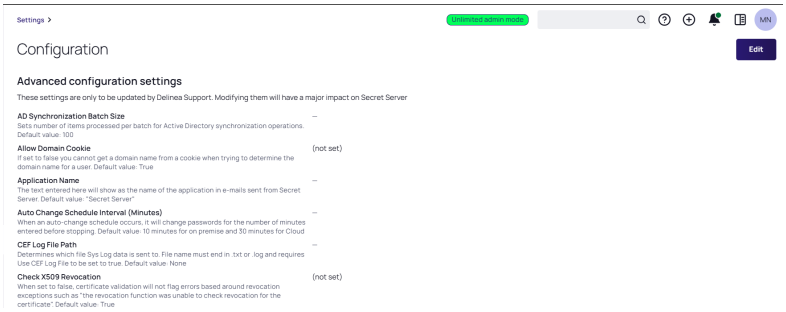
```
https://qa-test.acme-east.acmewidgets.com/app/#/admin/advanced-config-settings
```

The easiest way to get to the page is:

1. Open your Secret Server instance.
2. Navigate to **Admin > Configuration**. The (regular) Configuration page appears:

> Administrators in Secret Server do **not** automatically have access to all data stored in the system–access to data is still controlled by explicit permissions on that data.

1. Look at the URL for the page. The file name is `ConfigurationGeneral.aspx`.
2. Change the name to ConfigurationAdvance.aspx, leaving the rest of the URL as is.
3. Press **<Enter>** the (advanced) Configuration page appears:



4. Note the warning at the top of the page. It is serious, but it is also not *completely* correct. There are a few settings that may be important to your initial deployment. **Do not change any settings not directly discussed here without contacting Delinea Customer Service first**.
5. The following settings might need adjustment:
   - **IP Address Header:** If you are using a load balancer and multiple Secret Server Web server nodes, it is important to set this header to `X-Forwarded-For`. That way, user audits reflect individual user IP addresses and not your load balancer IP address.
   - **Secret Computer Matcher Once Per Discovery:** We mention this setting in the "Discovery Best Practices" on page 550 topic, where we recommend setting it to `yes` for large environment discovery. Otherwise, the matcher runs every five hours, regardless of how often discovery is configured to run.

Architectural and Design Considerations

You can view the Secret Server Architecture section. These reference architectures are, at minimum, refreshed every year and are created by our Professional Services Solutions Architect team. For this section, we provide some high-level architecture and design considerations that may help you design a more successful Secret Server or Secret Server Cloud installation.

> The following recommendations are primarily for Secret Server on-premises. For Secret Server Cloud customers, many of the recommendations are still relevant, even though you only have control over increasing distributed engines–the only Secret Server infrastructure you physically control when using Secret Server Cloud.

Consider some key questions about your SLA requirements for the application:

- What are the RPOs and RTOs for the application?
- Is high availability or disaster recovery required?

- Are you going to purchase Secret Server or Secret Server Cloud?

Answering these helps determine what initial infrastructure is needed for your environment. You can then look at the reference architectures to help select a variation of the reference architecture that works best for your requirements.


Many customers take a posted variation and alter it to meet their own needs.

When the Professional Services team works with our customers, we gather both architectural and stakeholder requirements to come up with a design that is sized correctly to meet all business needs. If you are planning to design Secret Server's architecture yourself, we suggest planning additional infrastructure based on feature utilization needs in the following order:

Session Recording

This is the most process- and memory-intensive feature of the product if it is used heavily. We recommend reviewing "Caveats and Recommendations" on page 1230 when planning to implement this feature, as it may require additional hardware or Web servers. Below are a few questions to ask yourself:

- Is the organization planning to use session recording and to what capacity?
- How many secrets may have session recording enabled?
- How many session recordings may occur concurrently?

We recommend only enabling session recording on secrets that absolutely need it–such as those with compliance or legal requirements. Otherwise, we recommend enabling session recording only on high value, high impact assets. This includes "global" admin accounts, domain administrator accounts, and other high-level privileged assets within your environment. This should minimize additional infrastructure just for session recording .

Discovery

This is another feature that can have a large impact on a Secret Server environment. A large enterprise discovering thousands of systems may require additional Web servers or distributed engines. Below are a few questions to ask yourself:

- How many systems do you intend to discover?
- How often should discovery run?
- How quickly does discovery need to complete?

We recommend using out-of-the-box discovery sources where possible. Since discovery cannot be scheduled to run at a specific time, consider enabling discovery for the first time during off-peak hours so it will run around the same time each day or week. If discovering a large number of systems, ensure you have ample Web servers and engines to handle the load. For example, increasing CPU count for each distributed engine can help distributed engines do more work in parallel.


Please see "Discovery Best Practices" on page 550 for details.

API Use Case

Employing multiple integrations with our product may impact a Secret Server environment. Below are a few questions to ask yourself:

- What integrations do you intend to use with Secret Server?
- What is the total number of API calls you anticipate per second, hour, or day?

We recommend that if you require several integrations with Secret Server where a high volume of API calls is anticipated, carefully consider how to configure your Web servers. You may want to have some Web servers dedicated to API use that have all Web roles explicitly disabled. You could place several such Secret Server Web servers in a load balancer configuration.

Remote Password Changes and Heartbeats

RPC and heartbeats may impact a Secret Server environment if used heavily. Below are a few questions to ask yourself:

- How many secrets RPC?
- How often should passwords be changed?
- How many RPC retries should be attempted?
- How often should we perform heartbeats?

We recommend carefully planning what types of secrets require different password changing schedules based on your company's information security policy. Generally, setting a large number of retry attempts for an RPC is not a good idea. The same goes for heartbeats. Match these settings to the business use case, such as 10 password retry attempts and having heartbeats occur once per day. These small refinements can greatly reduce the load on Secret Server. If you determine you need aggressive RPC and heartbeat schedules, consider having additional Web servers and distributed engines to handle the load.

Proxying

Heavy proxying can impact Secret Server infrastructure. Below are a few questions to ask yourself:

- How many systems are proxied?
- Are they SSH or RDP connections?
- What is the concurrent need?

We recommend proxy connections go through a distributed engine whenever possible. This offers a security advantage because ports, such as 3390 or 22, are not open inbound directly to your Web servers. You can review "SSH Proxy Configuration" on page 831 to size proxy requests.

General On-Premise Considerations

The areas mentioned below are often where we spend the most time with customers who have spent professional services time performing architectural health checks. These are the main areas we typically improve:

> For the highest scalability and reliability, Delinea recommends using RabbitMQ. MemoryMQ is an easier but less capable alternative and can be used for trials and proof of concepts but **should not** be used for production environments. Two exceptions are very small deployments and customers that do not use open-source software for compliance reasons.

- Ensure that you have a database maintenance plan in place for Secret Server. It should be implemented or reviewed by your organization's DBAs. Adjusting the data retention settings is not enough and does not substitute for having a maintenance plan.
  - Ensure that you have a maintenance plan in place to detect and re-build Indexes that are fragmented. This should be implemented or reviewed by your organization's DBA.

    > Note that your database maintenance plan may affect overall performance. See SQL Server Performance Improvement for more details.

- Ensure that RabbitMQ clustering configurations have work distribution policies in place. In most engagements, we use the AutomaticSyncMode policy.
- When designing multi-site, single-instance Secret Server implementations, be cautious when configuring Web servers and enabling roles on all Web server nodes when inter-location latency is high (50 ms or greater).
- When using the "local" site for Web processing when also using sites with distributed engine processing, consider using engine processing for the local site too. In most cases, when using both Web and engine processing, you are using both a built-in message broker (MemoryMQ) with (RabbitMQ).
- Consider having dedicated systems for Secret Server components, as proposed in our mid-range reference architectures. If using RabbitMQ, put it on a dedicated system that is not shared with the distributed engine service. Secret Server
- For large environments where discovery, RPC, and heartbeats will be used simultaneously, careful consider when to run discovery. Discovery can compete with RPC requests when both features are using the same site. For large environments, you may want to have a dedicated site and distributed engines for discovery and a separate dedicated site for secrets.

Securing the encryption.config File

Security is a process—not a product. Take a look at the "Security Hardening Guide" on page 1402 to ensure your implementation of Secret Server has optimal security. The guide contains more in-depth recommendations for not only configuring the application in a secure manner but also hardening the server or servers Secret Server is hosted on. That guide complements the information provided here.

One of the most important areas for Secret Server hardening is protecting the `encryption.config` file that is created during installation. After the product is installed, this file exists in the main `\SecretServer\` directory. It is a very important file. This file (unencrypted), along with a backup of your Secret Server database, is all you need to get a Secret Server environment back up and running. Thus, it is imperative that you protect it. There are two ways to protect the `encryption.config` for on-premises Secret Server and two others for Secret Server Cloud.

Secret Server On-Premises

For an on-premise installation of Secret Server, we recommend protecting your `encryption.config` file with an HSM. When using an HSM, though, there are other things that you should be mindful of:

- Is the HSM highly available?
- Is the HSM capable of handling a high volume of access requests?
- What methods are available for retrieving the key from a backup if my HSM were to crash?

> If your HSM is down and you do not have backups, there is nothing we can do to help recover your data. Carefully consider the configuration of an HSM for protecting `encryption.config`.

A second, less secure, option for protecting the `encryption.config` file is to use DPAPI combined with EFS. DPAPI is a setting that is enabled on each Web server within your Secret Server cluster. EFS adds an additional password to the `encryption.config` file. It is worth noting that both protection mechanisms can be compromised if an attacker were to log on interactively to Secret Server's Web servers and become a local administrator. Give careful consideration to securing remote access to Secret Server when leveraging DPAPI and EFS.

We recommend storing an unencrypted copy of the `encryption.config` file for disaster recovery scenarios where the Secret Server Web server is irrecoverable. Make a backup of this file immediately after installation (before securing it with a HSM or DPAPI + EFS) and to store the file on one or more media devices such as a hardware encrypted USB drive. The device should then be placed in a secure location, such as a safe. Access to the device should go through a chain of custody process in the event of an emergency where the original file is needed.

Secret Server Cloud

If you are using Secret Server Cloud, there are two main methods for protecting your `encryption.config` file:

- Delinea owns your `encryption.config` file and is responsible for keeping it secure. We put internal mechanisms in place to ensure that Delinea does not have access to your data without your explicit permission. Delinea
- You configure a connection to AWS KMS to protect the `encryption.config` file. The master key is stored in AWS and under your complete control, inaccessible to Delinea staff

See AWS Key Management in Secret Server Cloud for more details.

Privileged Account Management Strategy

It is important to have a privileged account management (PAM) strategy that helps you determine which types of features to leverage for your various accounts and sensitive data you will be storing. Below are some suggested guidelines for creating a strategic plan. We recommend reading all sections of the guide for a comprehensive look at ways you can secure your Secret Server. However, these guidelines will also link to other parts of the guide so you can choose to jump to a specific section for more detail about a particular topic.

Identify Data at Risk

Consider all the types of sensitive data your team needs to be securely stored and managed. Where are the biggest risks and pain points in your current password management strategy? Data at risk also often includes more than just passwords.

To get started, think through these key accounts and principles:

- All shared privileged accounts: these are accounts that don't identify an individual (for example: administrator, root, enable, service accounts). All of these should have randomized passwords that are changed frequently.
- Do your users have individual privileged accounts? Maybe each user has—separate AD account for domain admin rights?
- Every password in your organization should be different.
- Do your users have individual privileged accounts? Maybe each user has—separate AD account for domain admin rights?
- What passwords could be needed in an emergency, outside of regular business hours, or when someone is on vacation?

Typical account passwords and sensitive data being stored in Secret Server:

- Active Directory domain administrator accounts
- Active Directory service accounts
- Application passwords (such as SAP and, custom apps)
- Cloud Administrative or Privileged Accounts
- Database accounts (such as MS SQL, Oracle, or MySQL)
- Network equipment passwords (such as router, switches, firewalls, phones, and appliances)
- Sensitive files (such as private key files, SSL certificates, and network documentation info)
- Software license keys, serial numbers, personnel data, and Wi-Fi passwords
- UNIX, Linux, Mac root, and local user accounts
- Website passwords (cloud services, DNS, Amazon AWS, vendors)
- Windows local administrator accounts

Who Accesses Secret Server?

After determining the data you will store in Secret Server, the next step is to decide who will use Secret Server to access and manage that data. A common approach is to begin by focusing on one group of users and the passwords they use on a regular basis, later expanding to other teams once a good strategy has been put in place. However, you may find it more beneficial to organize Secret Server for use by all of your users/teams at once so you can design an effective overall folder and policy structure that will work well across all teams.

What Privilege Levels Are Necessary?

Giving a user access to an account in Secret Server can entail different levels of privilege. Do you want a user to be able to edit the username, machine, or password of a secret, or only view the secret? Should they be able to share the secret with other users? Once you incorporate use of the Launcher into your users' workflow for authenticating to an application, do they really need to know a password, or can you mask it? The Workflow Security section can help you determine and implement key measures to ensure users have least privilege necessary.

What are your Password Requirements?

It's unlikely that all your accounts will have the same password complexity requirements and rotation schedule. In fact, for best security, you should have some variation. You can create sets of password requirements to control password length, characters, and complexity, then apply those to various account types using secret templates. Secret templates also allow you to set a default expiration period, which can translate to how often an account password will be changed automatically.

Evaluate your Existing Setup

While transitioning to using a new tool for managing your passwords, it is important to take into account how accounts are currently used in your environment. The following questions can help evaluate this:

- Do some of your users have their own, individual AD domain admin accounts, or are there only a few shared domain accounts?
- Do users use local administrator accounts or privileged domain accounts for admin access to systems?
- Are permissions to resources (such as servers and applications) controlled using AD group policy?

Define Your Core PAM Strategy

There are a few different strategies that typically work best in Secret Server. Other methods of password management may work but require a more significant amount of time and effort to configure and maintain. The most commonly-used strategies are defined below.

Individual Privileged Domain Accounts

In this scenario, IT team members have their own domain admin accounts that are tied to their identity. They use these accounts to gain elevated privileges to resources such as production servers. Permissions to the various resources they're permitted to access are controlled by AD.

To implement this in Secret Server, each account is stored as its own AD account secret. Only the user tied to that account is granted permissions to the secret. A security setting such as check out (one-time password) or hide launcher password is enabled so the user depends upon Secret Server to use the account. Therefore, all access to that account will be audited. When the IT admin needs elevated privilege to a box, they check out or view the secret and then use the launcher to access the machine.

A benefit of this strategy is that there is not conflict with multiple users trying to use the same account for access to one machine. This strategy provides great accountability—the security team knows the exact user accessing an account and the machine being accessed. The password is not shared among multiple users, and all privileged access is audited by Secret Server.

A pitfall of this strategy can be that there is more management of permissions required in AD. While machines could be access or deny listed to force users to use the Secret Server launcher, thus controlling machine access through Secret Server, this can be tedious.

It is more secure, less work, and simpler to organize permissions for access to domain resources in AD. This strategy works best for organizations that already use AD heavily to control permissions of individual privileged users to domain resources. Ongoing maintenance will rely on updating permissions to resources in AD and ensuring that all new individuals' privileged accounts are being added for management under Secret Server.

Shared Privileged Domain Accounts

You may choose to have your users use shared privileged accounts to access resources. This strategy involves creating a few service accounts that have permissions to OUs or groups of computers. In Secret Server, these accounts can be limited with the Launcher so they can only be used to Launch to certain computers. This means you can limit the number of domain accounts created and set permissions more broadly (such as at OU level). These passwords could be changed on a schedule or, where possible, used with Check Out to change the password after each use. Using this setup, accounts can be designated for team or function and can have varying Check Out intervals set to ensure that only one person at a time is using each account.

A benefit of this strategy is if individuals do not already have their own privileged domain accounts in AD, then giving them access to shared accounts means less setup in AD while still maintaining accountability for who uses which account, and which machine they access.

A pitfall of this strategy can be that if the team (or function-specific accounts) cover a broad number of machines that can be accessed, it may be a lot of work to set up launcher allow/deny lists to control access through Secret Server. However, if these permissions are set only through AD, it will be difficult to have the visibility into these limitations for an auditor.

Hybrid of Individual and Shared Accounts

Sometimes, your employees' roles may require longer, more specialized access. For those accounts, you can have individual privileged domain accounts, and for the other regular users you can use a few shared privileged domain accounts. All of these can be stored in Secret Server, but with different settings governing their usage. For example, the shared accounts would still have check out enabled, while the individual privileged accounts will simply have permissions limited to an individual user, possibly with the password hidden using hide launcher password.

What Is the Highest Risk?

Implementing a comprehensive PAM policy should eventually cover all of your privileged/shared accounts, but this can take some time. When looking at where to start, it is important to consider the areas of risk–where are the areas that need more immediate attention:

- Is it local Windows admin accounts all sharing the same password?
- Pass-the-hash vulnerability?
- Protecting your network equipment passwords?
- Avoiding fines for not meeting compliance mandates?
- Password misuse and auditing employee access to accounts?

Choose a starting point that gives your organization the most value, and then branch out from there.

Users and Groups

At minimum, the administrators who manage and use your organization's privileged passwords and data on a regular basis will need to access your Secret Server. Secret Server users can be defined in a few ways:

- Active Directory user accounts.
- Local Secret Server user accounts.

- User accounts from an Azure Active Directory tenant.
- User accounts from another OLDAP source (Basic/Kerberos).
- User accounts from SAML integration (often AD accounts). If local accounts are provisioned via SAML, they must correspond and match local user accounts that are within Secret Server.

  SS also has the concept of groups, which can be local (you create them in SS), AD-synced (security groups from AD), oLDAP groups, or AzureAD groups . Groups are a powerful tool for assigning and maintaining permissions to secrets, and therefore should be given careful thought and planning. Below we review the two most common account and group strategies our customers use. These same concepts can apply for other directory service accounts and groups other than Active Directory.

Local Secret Server Accounts

Local users and groups have to be created and managed manually in Secret Server, as they are not integrated with AD. The first account you create in Secret Server is an example of a local account. Local groups can include local users and AD accounts, and can have a user established as the group owner that is permitted to add or remove users to or from the group.

Active Directory Accounts

AD accounts can be added for access to Secret Server either manually (one by one) or by AD security group. When adding users by security group, you choose which groups Secret Server will synchronize with AD to update which users' access to Secret Server is enabled or disabled. AD group synchronization happens on a regular, customizable interval to keep group membership changes that happen in AD up-to-date in Secret Server as well.

Local or Active Directory Accounts?

We recommend using one of these options:

- Only local users and groups (best security)
- Only AD users and groups (most convenient)
- A hybrid of AD users and local groups (balance of security and convenience)

You need to choose an option that provides the levels of security and convenience that are acceptable for your organization. Using the AD accounts option is easy for user maintenance, but it limits the security of Secret Server to the level of security of your AD. This may be fine—just be sure to consider the question of domain admin access to AD in combination with Secret Server permissions.

Only Local Users and Groups

Creating local users and groups within Secret Server provides a lot of flexibility because you can tailor permission assignment by group to your exact needs. The major benefit of local users and groups is security: users and group membership can be controlled entirely by role-based access control (RBAC) within Secret Server. However, this approach requires more maintenance because creating or deleting users and managing group membership has to be controlled in Secret Server.

Only AD Users and Groups

If you are considering using AD users and groups for Secret Server access and permissions assignment, review your teams that need access to Secret Server. Compare them to the corresponding groups in your AD. If your AD groups map to ways you want to assign access to secrets, you can synchronize your AD groups with Secret Server and start assigning permissions to secrets (and levels of those permissions—View/Edit/Owner) by group. You can then effectively manage Secret Server access and secret permissions completely from AD by changing AD group membership.

Many customers choose this option because they can maintain control in AD and do not have to worry about any user or group maintenance within Secret Server. If you want to use this option but your AD groups don't match the way you want to assign secret permissions, you will need to create new AD groups to match this, or may want to consider the hybrid approach (below), using local groups instead.

This method, while more convenient, may require additional considerations:

- How are these AD groups being protected?
- Are there controls in place which require elevated or high privilege accounts to modify these AD groups?
- Are there alerts in place for when these groups are modified?
- Is the information security team closely monitoring these groups?

Hybrid of AD Users and Local Groups

A third option is to create local groups in Secret Server and add AD users to those groups for the purpose of organizing how permissions are assigned to secrets. Many customers who use this setup will create a single AD security group (for example, SecretServerUsers) to use to synchronize their AD users with Secret Server for log on. They then create additional local groups for their users to, which gives them permissions within Secret Server, such as to their teams folder. They may also be added to other Secret Server groups that provide them with other privileges within the environment.

This approach is more secure than using only AD groups and users, but if Active Directory were compromised, intruders may still be able to reset an account password and gain log in access to Secret Server. If secrets are stored in that user's personal folder, those secrets may be compromised which may lead to lateral movement elsewhere within the organization.

Business Users

A "business user” is a non-information-technology user. Business users typically use Secret Server for vaulting credentials for access to departmental websites, applications, and personal accounts in an enterprise-grade secure vault.

At Delinea, we strive to ensure that all Secret Server Enterprise Vault users get the maximum value possible from the product. Business users do not need all vault features, but they can benefit by securing their passwords and credentials to prevent abuse or malicious use.

Business users can:

- Access secrets: They can create, update, and delete their own secrets within Secret Server. For example, a user signing up for an online service can use our password generator to create a strong password, store that

password in Secret Server, and later use Web Password Filler to access it when logging on.

- Request and approve access to secrets: Non-privileged secrets may need approval workflows. Business users can request access to these secrets and can approve access for others. For example, if access to an organization's social media accounts required authorization from a member of the marketing team, a business user could request access to the secret, and another business user could then approve access.
- Share secrets with other users: Business users can share non-privileged secrets with other users of Secret Server, either IT or business users.
- Access secrets using our mobile app: Business users can use the Delinea mobile application to access and manage their secrets.
- Use Web Password Filler: Business users can use our Web Password Filter (WPF) for accessing websites and auto-filling credentials from their secrets.
- View audits of their secrets: Business users can use auditing to see who accessed their shared secrets.

A "non-privileged secret" refers to any secret stored within Secret Server that does not grant elevated access rights or permissions. The most common type is regular user-account passwords.

Table: Allowed Actions for Business and IT Users

Action	Business User	IT User
Approve access to secrets	•	•
Configure password rotation		•
Configure security features for a secret		•
Create, delete, modify folders and add secrets to any folder		•
Create and delete personal folders	•	•
Create, update, and delete personal secrets	•	•
Create and manage Integrations, workflows, pipelines, discovery, sites, distributed engines, HA/DR, and more		•
Launch secrets they have access to	•	•
Request access to secrets	•	•
Share secrets	•	•
Use any administrative functions of Secret Server		•
Use any launcher including RAS (Platform)		•

Action	Business User	IT User
Use Connection Manager to launch RDP or SSH sessions		•
Use the Delinea mobile app for access to secrets	•	•
Use the Secret Server SDK or API		•
Use Web Password Filler	For Non-IT-related tasks, websites or portals	For any tasks
Use Web templates in Secret Server	For Non-IT-related tasks, websites or portals	For any tasks
Utilize any templates including custom templates		•
View secret audits for their own secrets	•	•
View user audits for their own secrets	•	•

Administrators can restrict template visibility for Business Users. By default, all users have access to all templates, but business users typically need access to the following templates:

- Bank Account
- Combination Lock
- Contact
- Credit Card
- Healthcare
- Password
- PIN
- Security Alarm Code
- Social Security Number
- Web Password (Non-IT-related)

Authentication Strategy

Defining your authentication strategy ensures you have standardized authentication practices in line with your Secret Server RBAC scheme. Secret Server offers a wide variety of authentication options that can add flexibility and security to your business user's authentication process.

Strong Authentication

Protect the tool you are using to secure your privileged accounts by adding a second factor of authentication for users logging into Secret Server. Two-factor authentication can be added whether users are logging in with local or AD accounts. For more information about using two-factor authentication with Secret Server, see the "Security Hardening Guide" on page 1402.

SAML

If your organization is already using SAML for SSO across your organization, it might be a good option for Secret Server authentication too. SAML uses browser-based communication, between the service provider (Secret Server) and the identity provider (SSO providers) to broker authentication. For more information on configuring SAML with Secret Server, please see the . The major benefits SAML provides are:

- A consistent MFA strategy across all applications in your environment.
- Simplified authentication communication: The browser handles the process. For SSC, if the authentication strategy is to authenticate against the domain, that communication must flow from Secret Server Cloud to the distribute engine and finally to the domain controller and back for authentication. SAML shortcuts this by having the browser communicate to the service provider and identity providers, reducing authentication latency.
- Easy to configure, manage, and add new users.
- Supports multiple MFA options based on conditional access. For example, a user may only need to verify with one factor for accessing less critical apps, but Secret Server uses two-factor authentication.

Directory Services

Secret Server provides a multitude of authentication options through directory services. It can sync users into the application from various LDAP sources. It is important to use an outside authentication source to automate user provisioning. The User Account Options setting in the directory services configuration provides these options:

- Users are Enabled By Default (Manual)
- Users are Disabled By Default (Manual)
- User Status Mirrors Active Directory (Automatic)

There are benefits to each strategy, but the last one is usually best. Using a hybrid group structure prevents new users from gaining permissions before they have been reviewed, and if they are disabled in active directory, they are automatically deprovisioned from the system. This provides automatic user management and easier offboarding strategies.

If you decide to use a manual strategy, we suggest using the automatic user management feature to disable users who have been inactive for a defined period of months. This can prevent long-inactive accounts from being compromised and keep your list of active users current.

Roles

Roles control which features of Secret Server a user is able to use, view, or administer. Existing roles can be customized, and new roles can be created as needed. Secret Server comes with several roles by default, including administrator, user, and read only. You should review the default roles and decide whether your organization needs further roles for various purposes such as third-party consultants or auditors.

> Users with the default administrator role (which contains all role permissions available) do **not** automatically have access to all data stored in your Secret Server. secrets are only visible to a user based on the explicit secret permissions assigned to them.

We strongly recommend pulling one or both role permissions pertaining to unlimited administration mode out of the default administrator role. Unlimited administrator mode is a "break-the-glass" feature that allows a user to view all secrets in Secret Server. By splitting the unlimited administration permissions into separate roles, it ensures no one user can both turn on the feature and operate as the unlimited administrator.

Commonly, operational employees are assigned to the "unlimited administrator" role and a CISO or senior manager that is responsible for Secret Server is assigned the "configure unlimited administrator" role.

> For more information about how unlimited administration mode works and how to effectively control the relevant role permissions, see the "Security Hardening Guide" on page 1402 and "Unlimited Administration Mode" on page 273.

Other sensitive roles you may want to directly assign to individuals include:

- Administer Role Assignment
- Administer Role Permissions
- Bypass SAML Login
- Create Root Folders
- Delete Secret

For administrator-related roles, we typically recommend having these associated to your planned internal Secret Server subject matter experts. Usually customers have a dedicated group or groups associated with administrator only roles and functions. You can create administrator tiers depending on the size of your organization and the tasks you expect the administrators to perform. This can reduce administrative overhead and provide a path for employees to gain further experience with the product.

We recommend creating AD groups for these administrative roles to ensure that these groups are protected and can only be modified by higher-privilege accounts. Ensure proper monitoring and alerting is in place when creating groups that are intended for high-privilege role access within Secret Server.

Role Definition and Assignment

Once you have defined your roles, they will seldom need to be changed. Access to modify and assign roles should be tightly controlled.

Group Assignment

If roles are assigned to groups, then assignment of the groups will also need to be controlled. Often very sensitive role permissions, such as unlimited administrator, are assigned at the user level to limit the risk of granting group assignment permissions. Roles that are individually assigned should be routinely audited at least once a year to ensure users are only assigned the permissions needed for their job. See "Unlimited Administration Mode" on page 273.

Permissions

You have different sets of passwords that should only be viewed by particular administrators. You may also have passwords that should be read-only to some administrators, editable by others, and not even visible to still others. All of these options are possible using the permissions within Secret Server.

Permissions can be allocated at the individual user level but it tends to be easier to manage over time if you allocate your permissions at the group level. You will need to decide if your existing AD groups could work for these permissions or if you need to create new AD groups or if you want to create and manage local groups in Secret Server.

For more information about what each level of permissions entails, see the "Secret Server Role Permissions List" on page 1257.

Folder Structure

Using Folders to Control Access (Inherit Permission)

You can apply permissions (View/Edit/Owner) at the secret level. This allows you to apply very granular permissions on a single secret if needed. Managing permissions on each secret is powerful for situations where you need that flexibility, but it tends to be harder to manage over hundreds or thousands of secrets. Instead, you should consider using folders to control permissions for most secrets. This can be done by creating a folder structure that best represents your organization, teams or data being stored and then applying permissions on the folders, using inheritance across folders where appropriate. Secrets placed in a folder can then inherit the permissions of the folder.

Deciding on your Folder Structure

The folder structure creates a hierarchy for organization and permissions. This means that folders near the root level need to break out access in high level terms and then get more specific permissions (typically breaking inheritance) as you move down to the "leaf level" sub-folders.

For example:

– Customers
– Human Resources
– Information Technology
—– Development Services
—— Programmers
—– Technical Services
—— Database
——— Oracle
——— SQL Server
—— Systems
——— Network Infrastructure
——— Unix
——— Windows
– Vendors

The most typical configuration is to break out the folders based on the teams that need to use those folders with the most restrictive permissions at the deepest subfolders of the tree.

For instance, an Oracle DBA might have the following permissions on the above folders:

– Information Technology (view)
— Database (view)
—— Oracle (view/edit/owner)
—— SQL Server (view/edit)
— Technical Services (view)

> If the "require view permission on a specific folder for visibility" setting (Admin > Configuration > Folders) is enabled, a user cannot see the full folder structure unless they have view permissions on all the parent folders of a folder. For example, a user with view on the Oracle folder in our example, would also need view on Database, Technical Services, and Information Technology to see the full folder path.

There are settings under **Admin** > **Configuration** > **Folders** to control whether inheritance on folders and secrets should be turned on and also whether users should always see all folders. There are many ways to configure this for your organization.

The most common approach is:

- Use inheritance.
- Do not allow users to see folders unless they explicitly have view permission by enabling the "require view permission on a specific folder for visibility" setting.
- Require all secrets to have a folder.

> Consider using our "User Teams Overview" on page 1284 feature to align your groups within Secret Server to a team. This can help prevent users from sharing secrets with other individuals outside of their own team.

This approach allows different teams or even different departments within your organization to use the same Secret Server instance independently.

If a business need arises to break permission inheritance on a folder or secret, we recommend tracking or auditing those folders because manually applying permissions can increase your administrative overhead.

Secret Policy

A *secret policy* is a set of security and remote password changing settings that are normally applied to a secret on the Security or Remote Password Changing tabs. The benefit of using a secret policy is not only that settings can be applied in bulk to secrets (that is, by folder), but that these settings can also be enforced, preventing users from changing them.

Secret policies should be established to apply settings to secrets that are key to the workflow your organization is working toward. For example, if your primary concern is more detailed auditing around service account usage and you also have a requirement that all service account passwords change every 60 days at 2 A.M. on the next Tuesday, you can create a policy that includes these settings and apply it to the folders that will contain all of your service accounts. Whenever new accounts are added to the folder, such as when they are imported via discovery, the settings will automatically be applied and enforced.

Secret policies can also be updated after they have been assigned to folders. Therefore, if your password policy changes and you need your service account passwords to change every 30 days, you can update the policy and it will immediately apply to all secrets the policy is assigned to.

As with permissions, secret policies can be inherited too. Be mindful of where you disable secret policy inheritance to ensure that exceptions to secret policies are tracked. Also, disabling secret policy inheritance may lead to increased administrative overhead.

Discovery

This section discusses some key best practices around using Secret Server's Discovery feature to find and manage accounts in your environment.

Discovery Workflow

While it may be tempting to immediately get started using discovery to get your accounts under control, there are a few things you can do ahead of time to make the enforcement of your organization's password policies more streamlined:

- Know which secret template you want to import accounts to. This can effect password changing and Launcher settings that are applied to your imported accounts.
- Have a folder structure established so you have folders appropriated for each type or category of discovered accounts.
- Apply a secret policy to the folders you import to.

Having these settings in place can save you the considerable amount of time it could take to have to re-organize all of your accounts and policies post-import.

Enterprise Deployment Considerations

We broadly recommend starting small and choosing specific objectives when working with discovery. If you are an organization that has 15 domains, for example, you may choose to first work with discovery within the domain you are most concerned about. Make the objectives even more specific where possible. An example first objective might be to configure discovery for finding all local administrator accounts on all your servers and creating discovery rules for ensuring that new servers have their password changed shortly after being built. Systems with internal elevated risk may also be a good place to start. Other examples are provided below.

Cloud Accounts

In more recent Secret Server versions, we support discovery of "Google Cloud Platform Discovery" on page 613 service accounts, VM instances, and "AWS Account Discovery" on page 599.

Local Windows Accounts

How many local Windows accounts in your environment use the same password? Are they local admin accounts? Use discovery to quickly mitigate the risk of pass-the-hash attacks by finding all of your local Windows accounts and setting their passwords to unique, strong passwords managed by Secret Server. Where your admins previously had to remember one password to access all machines with local admin rights, they now have to remember zero passwords because they can use Secret Server to find the machine and launch an RDP session using the local admin account without ever knowing, copying, or typing the password.

Find Backdoor Accounts

Ensuring that users are not creating backdoor administrative accounts on Windows machines is very important as these can compromise general security as well as open the potential for a user to access a machine directly without being audited. By running discovery on a regular interval and having discovery rules alerting you when new accounts are found, you can ensure that users any new local Windows account being created are identified in addition to being either removed or brought into Secret Server.

Service Accounts

Many organizations do not know where their AD service accounts are being used across the network. By using discovery to scan your network, you can find all of the Windows services, application pools and scheduled tasks that are run by AD service accounts. Once these accounts are found and brought into Secret Server, having discovery run on a regular basis will find any new locations where the account is being used since they were added to Secret Server. With discovery rules, those additional dependencies can be automatically added to the existing secrets. We recommend making sure that the service account discovery has run before using Secret Server to change the service account password.

Unix Accounts

When scanning for Unix accounts, we recommend using SSH key validation, as discussed in the "Security Hardening Guide" on page 1402. This ensures that you are only connecting and trying to authenticate to UNIX servers that have a valid and trusted SSH key.

ESX/ESXi accounts

Local accounts on ESX/ESXi systems should not change once the server is set up and configured. We recommend creating discovery rules that monitor your ESX/ESXi servers and email the proper teams to inform them of any new account found. These accounts really should not be created, so it is important to monitor them and ensure that no one is creating them maliciously.

Workflow Security

Often you will have situations in which you want users to have access to accounts, but only under certain circumstances, such as on a specific day or after the approval of a manager. Maybe your compliance requires that you have the ability to monitor an active RDP, or that you use a one-time password for certain accounts. This section examines best practices around workflow security settings in Secret Server as well as scenarios when these settings are commonly used.

Hide Launcher Password

Many times, giving an employee access to a resource through Secret Server does not require that he or she have access to the actual password for the account used. As long as the application a user needs can be started by the launcher, there is no reason the user needs to copy/paste or type the password. The hide launcher password setting implements the following:

- Users with access to the secret will see only asterisks (****) in the password field
- There will be no copy-to-clipboard, field history, or unmask icons next to the field

> Users with edit permissions to a secret with "hide launcher password" enabled can still view the password when editing the secret. To prevent all possible access to the password, limit users to view permission only.

This can be an important way to reduce exposure of your privileged account passwords. Hiding launcher passwords can be enabled for secrets under the Security tab of a secret or by applying a secret policy. You can also remove the ability for a user to see the password for any secret with a launcher by removing the "view launcher password" permission from their role.

Require Approval

The "requires approval for access" setting is typically employed in the following cases:

Simple approval workflows:

- When a user should be required to request access to a secret for a certain time period
- When an administrator would like to approve a user's access to a secret in advance for a time in the future (such as a maintenance period outside normal business hours)
- When a group of administrators would not like anyone to access a secret without the approval of another administrator

Advanced approval workflows:

- When requiring a multi-tier approval process that involves having more than one individual approve access to a secret
- When requiring multiple workflow steps, each with different reviewers and a varied number of required approvers
- When selecting "owners" as a review group

This setting can be turned on under the Security tab for an individual secret, but can also be applied via a secret policy. When enabling "requires approval for access," remember that users will still need to have at least view permission to the secret to request access to it. Once access has been granted to the secret, they have whichever level of permission was assigned to them for the secret (view, edit or owner). The approvers of the secret are specified when enabling the setting, and these individuals will be able to modify the time that the requestor originally submitted their access request for or deny the request altogether.

> Please see our documentation on "Workflow Overview" on page 1089 for more advanced workflow use cases. For highly sensitive or privileged accounts, we do recommend implementing multi-tier approval processes where possible.

To require all approvers of a secret to also request access from another approver, be sure to enable the "owners and approvers also require approval" setting.

Require Comments

Requiring comments to be entered when viewing a secret can be an excellent way to ensure users are accessing a secret for legitimate reasons. You can even view the comments in the audit of the secret to historically track if a

secret was accessed for the originally intended purpose. Managers can routinely review these comments and determine where employee training may be required.

A common example would be enabling require comments on a domain administrator account that is stored in Secret Server: A user may enter a comment that indicates he or she needs to use the domain administrator account to "perform adding a user to a group." In many cases, a domain administrator account should not be used for this purpose and often this work can be done with a lesser privileged account within the environment.

Requiring comments can also be combined with "Ticket System Integration" on page 222. This is a great way to align secret access with a valid ticket number and a comment. This can help with compliance and track usage of a secret tied to a specific task, which may provide more granular information as to why a secret is needed.

Check Out

There are times when users need to be able to access a password directly, but you still want to have control over how long they are able to use the account without the need to approve access each time. In this case, hiding the launcher password is not a possibility, but there is also concern about having the user know what the password is after they are done using it. Another concern if often the risk of the hash of the password being stored locally on remote devices after each use and potentially being vulnerable to a pass-the-hash attack.

"Check out" is a security setting that means:

- Only one user at a time has access to a secret
- A user can only access the secret for a predetermined check out interval, such as 30 minutes
- At the end of a check out interval (check in), or when a user manually checks in the account before the time is up, the secret is available for check out by another user
- When enabled, the password can also be changed automatically upon check in

Domain administrator accounts are a great example of a case in which using check out to change the password every time it is used can be extremely beneficial. This ensures that users are not copying the password to Notepad or writing it down for later use and also invalidates the hash that was stored on the remote machine after a remote desktop session.

Check out can be turned on under the Security tab for an individual secret, but can also be applied via a secret policy.

Session Monitoring

For critical systems and highly privileged accounts, sometimes simply having an audit trail showing when someone viewed the account in Secret Server is not enough. Maybe the auditor also wants to be able to review what was done with the account on a remote session. For these critical secrets, it is recommended to enable session recording for the secret. When session recording is enabled, all launcher sessions can be recorded for later viewing by the auditor or manager in the event they need to investigate the actions performed during a remote session.

What constitutes a "critical system" is subjective. Departments may define this differently, so having them involved in those discussions can be helpful. Some of these systems may be explicitly selected based on risk, compliance, or from the auditing team.

One of the most important things is to not consider all accounts or secrets critical, enabling session recording for them all. That can cause performance issues within your environment—session recording is the single most intensive feature of Secret Server.

Before enabling session recording, you may want to evaluate your users' roles to determine who can monitor real-time sessions and view recordings. The permissions to look for are "administer session monitoring," "view session monitoring," and "view session recording."

Session monitoring can be turned on under the Security tab for an individual secret, but can also be applied via a secret policy.

Secret Templates

Secret templates in Secret Server define the types of data (secrets) that can be stored, and the settings for that data. You can store just about any type of sensitive data in Secret Server.

It is important to review the available templates and decide which ones should be available to your users as well as where you would like to make changes to the default templates included.

Configuring Templates

You can customize existing templates or create new templates if necessary. Many templates are included by default that cover common account types. For example, the AD Account template contains the following settings:

- Domain, username, password, and notes fields
- 30-day expiration period, applying to the password field
- RDP launcher, requiring user input for computer to connect to
- Password changing and heartbeat enabled
- AD password changer, with default password requirements

These settings are typically sufficient for most organizations to use out-of-box. However, you may wish to enable other settings or change settings such as enforcement of a naming convention or more complex password requirements. In this case, you have the flexibility to either modify the existing template, copy the existing template to use as a base for a new template, or create a new template from scratch. The following sections cover some fundamental template settings available for you to customize.

One best practice we often recommend is simply leaving default templates the way they are and duplicating the templates you plan to use. Then customize the newly duplicated templates as needed. Name them something your employees will recognize and readily use.

File Attachments

Do not forget files. You can have fields on your secret template attaching files. This can be used for storing license key files, private keys, SSL certificates, even Microsoft Word or Excel documents that contain sensitive data.

Naming Patterns

Secret Server supports enforcement of naming patterns for secret names. Naming patterns allow you to maintain consistency for secret names and can help ease both browsing and grouping secrets by name. Naming patterns use regular expressions and allow you to enter a descriptive error message to describe your naming standard to users. The most common naming standard used is RESOURCE\ACCOUNT (for example, server0001\administrator). You can find this setting by clicking Edit from the template designer page.

Password History

Secret Server automatically keeps all history on all fields on a secret template. This means that all previous values for machine, username, password and any other fields will be kept. This is helps ensure that previous passwords can be found if needed.

Password Requirements

Password Requirements determine the password compliance rules (for example, 16 characters, one uppercase, one lowercase, one symbol and one number). These can be customized and applied to passwords at the secret template level or per individual secret (under the Security tab). This controls the complexity of passwords generated by Secret Server. Password requirements can also be enforced when users try to edit or create new passwords, and can be viewed for password compliance in reports. This allows you to have different complexity rules for different types of passwords if needed (such as Oracle, SQL, Windows, and UNIX). You can choose to have Secret Server enforce the password requirements on add/edit by turning on validation on the secret template (click Edit from the template designer page).

We have added new password "Template Password Requirements" on page 1181 to recent versions of Secret Server, which further helps create unique passwords. We also added "Creating and Editing Custom Password-Exclusion Dictionaries" on page 1166 that can help personalize which words may never be used as part of a password that is generated.

Talk to your security management, auditors and industry experts to find out the best password complexity settings for your environment. Do not hesitate to stipulate complex passwords, such as 100-character random-generated passwords with symbols, alphanumeric uppercase and lowercase)—using a platform like Secret Server makes it easy to work with passwords so complexity and length do not matter (for launchers, copy-to-clipboard, and auto change). In fact, very large passwords can add to security since administrators will be far less likely to remember them or write them down or want to type them.

Another thing to consider when creating password requirements is which character sets should be used. Some systems may not work well with certain characters. For example, underscores can be problematic in certain mainframe environments. You can create your own character sets (Admin > Secret Templates > Character Sets) for use in password requirements. These can then be used when passwords are generated by Secret Server.

Secret Expiration

Secret Server uses expiration to ensure that passwords are changed on a regular basis. Secrets can be set to expire on an interval such as 30 days (or other intervals as needed). Expiration is often combined with automatic password changing to control how often a password is changed (whenever it expires, Secret Server will queue the secret up for a password change).

You can also control which field is used for expiration. This does not have to be the password field—you could use expiration on a license key and set expiration to when the license is going to expire. When a secret expires, you can then update the expiration field (say license key) and it will no longer be expired. This is a generic way to ensure that a specific field on a secret is changed on a regular basis.

Session Launcher

The Launcher can be configured on the secret template to allow any tool to be launched using the secret such as Remote Desktop, PuTTY, Web launcher or a custom launcher you configure for a particular executable file, for

example, MS SQL Management Studio, SSH clients, FTP tools and more. This can also be used with the "hide launcher password" setting to allow administrators to launch tools without revealing the password.

Template Management

It is worth spending time in the beginning to get your secret templates the way you want them before users start adding data. Therefore, when a user goes to create a new secret it will be clear which secret template to use instead of selecting the wrong one and attempting to fit account information into an unsuitable template. You can use an option on the secret called "convert template" to later convert a secret to another template, but it is much simpler to plan before your organization begins adding data.

Basic Configuration

When creating new secret templates, make sure you configure Remote Password Changing, password requirements, secret expiration and the launcher. Ensure your secret template names are descriptive and use terms your users understand. For instance, if you have one template that expires and one that does not, make sure it is clear from the name. If your organization does not use the term AD account, change it to match the organization's language.

Deactivate Unused or Retired Templates

Secret Server comes with many secret templates preconfigured. You should decide which you want to use and then deactivate the others. You can also retire secret templates if your requirements change over time—secrets remain when a secret template is deactivated but no one will be able to create new secrets for that secret template. Secret Server SS uses soft deletes rather than hard deletes (data is marked as inactive rather than actually deleted), which is essential for auditing. Secrets and secret templates can be inactivated but not deleted.

Limit Secret Template Administrators

Changing secret templates should be limited to only a small subset of your Secret Server admins. Create a separate role that has the "administer secret templates" role permission and remove it from administrator if you have a lot of administrators. Once you have secret templates configured, it is unlikely they will need to be changed frequently so very few people should need access.

Override Settings at the Secret

Many of the settings at the secret template can also be overridden at a secret based on that template. For example, if you create a secret for your AD service accounts with a 30-day expiration but need 90 days for a specific AD service account, you can set it to 90 days for that one secret. This gives some flexibility for secrets that need to behave differently than other secrets using the same secret template.

Alerting and Reporting

Event subscriptions are a great way to send alerts based on various activities. One of the most common event subscriptions we recommend is alerts based on "unlimited administrator" mode being turned on. This can be aligned to alert your CISO or a manager within your information security team, as this should happen only rarely.

See "Unlimited Administration Mode" on page 273 for details.

Other useful event subscriptions to consider:

- Backup Configuration – Backup Failure
- Configuration – Edit
- Encryption – Key Management Disabled
- Role
- Role Permission
- Site – Engine Offline

These alerts can be sent to different people or can even be sent to users that do not have a Secret Server account. For some of the suggestions above, you may have some of these alerts go to a manager of the systems administration team rather than the information security team. Sending alerts to team members that are responsible for different portions of the application allows for flexibility in who may have to respond to these events.

There are many useful built-in reports. For example, a license audit report may be useful to an auditor, while the built-in reports for secret policies may be useful for information security. It is a good idea to meet with various teams to determine what reporting requirements they may have. Please review our "Creating Event Subscriptions" on page 311 and "Scheduled Reports" on page 896 documentation for more information.

Data Retention and Database Size Management

The "Audit Data Retention" on page 278 documentation may be helpful for very large environments where there is audit and log retention flexibility. For example, some of these tables within the database are sizeable, so if your environment exceeds 50,000+ secrets, it may be a good idea to make some adjustments. Similarly, you could have a smaller environment, say 25,000+ secrets, but if you are using all Secret Server features heavily, adjustment might be helpful.

Deleting the data within here should not substitute for a database maintenance plan and should be only considered complimentary to one. Lastly, individual audit tables, such as the "secret audit", cannot be managed independently outside of this configuration unless adjusting those tables directly within the database. Thus, we do not recommend doing unless under direct instruction of our support or professional services teams.

API and Extensibility

APIs and built-in extensibility features are one of the best ways to improve the automation and flexibility of Secret Server and address novel use cases. PAM Maturity relies on the consistent handling of credentials across your organization. Using the API is a great way to enforce consistency, without additional administrative overhead.

Features like event pipelines can help to organize secrets and users, as well as tweak some settings that are not enforced via policy. Additionally, you can discover and manage new device types using extensible discovery or custom password changers. This section covers some high-level best practices for these features and how best to employ them in your environment.

Running PowerShell with Secret Server

Secret Server's extensible features almost all use PowerShell from the Web servers or distributed engines to execute code. Create a service account for these tasks and store it in a secure location with the other Secret Server service accounts. Documentation for our "APIs and Scripting" on page 1466 components can be found in our Knowledge Base.

PowerShell Runspaces

Runspaces are instances of the PowerShell engine within a process. They define the context a PowerShell runs in and preserve session state using variables and functions that you define, including modules that you load.

For scripts executing on the local or default site, Secret Server generates a runspace to localhost on the Web server (Web site processing) or distributed engine (engine processing or Secret Server Cloud). For scripts running against a specified site, any engine associated to that site may run the script and accomplish the work. This means any dependent configurations need to be made across all Web servers or engines within a site.

This runspace is generated using a specified secret credential in all cases. The credentials used are critical in the operations of PowerShell in Secret Server and a least-permissioned approach should be used.

A domain account, that is, a member of the "remote management users" or "local admin" groups, on the engines or Web servers will have enough permissions to generate the runspace Secret Server needs to execute code. Further restrictions to deny interactive log on can be applied; however, it is important to know what account to use for a least-permissions approach.

You can simplify the assignment of the privileged account in individual locations in Secret Server by specifying a "site run as" secret. This is a default secret for PowerShell running that prevents you from having to manually associate the secret in each place it is used. We highly recommend to configuring this setting on each site under the site's settings.

CredSSP

CredSSP is an authentication mechanism that is designed to delegate credentials across multiple sessions in PowerShell. By default, the runspace uses CredSSP authentication. This is not ideal as there are some security concerns with CredSSP. It is generally better to disable CredSSP authentication under all used sites, and instead implicitly pass credentials into the scripts. CredSSP does have dependent configurations that are required before it can be used.

However, If CredSSP is required, be very explicit with your delegate computer list. Allowlisting this option with an asterisk (*) may seem easy, but it allows an attacker to scale horizontally across your network.

API

The API is a powerful tool for improving automation and flexibility of your Secret Server deployment. Automation tools can pull dynamic credentials from the vault or leverage the API to create new secrets as part of an automation pipeline.

If your organization does not have script experts to improve your PAM program's flexibility, Delinea Professional Services can help to scope and create custom integrations to address your use cases. Contact your account manager or customer success manager to engage professional services.

API Authentication

Carefully consider your authentication strategy for the API when automating workflows. Authentication strategies that do not require hard coding a password into a configuration or script file are preferred. Secret Server Provides a few mechanisms to accomplish this:

Software Development Kit

The Secret Server Software Development Kit (SDK) is a mechanism to authenticate machines to a Secret Server instance, without having to pass implicit credentials into the system for each set of calls. To ensure proper RBAC, the SDK is associated to a specific API user in Secret Server. On the local system, once the SDK is initialized, it leverages DPAPI to encrypt the config files, tying it to the user who initialized it. You can use multiple SDK instances on the same machine; however, each instance needs its own directory.

Integrated Windows Authentication

You can enable Integrated Windows Authentication (IWA) in IIS to use native Windows authentication for API user authentication. This allows the domain to supply authentication between the Web server and Secret Server. This is a great option to remove credentials and automate authentication for automation processes in your environment.

> IWA is only available for Secret Server On-Premises.

Event Pipelines

"Event Pipelines" on page 285 are a powerful feature that configures tasks that run based on triggers and filters. If scripting is not an option for you, pipelines can be a script-free way to accomplish a lot of what the API provides. You can use event pipelines to build the core automation of a mature PAM deployment.

Distributed Engine and Protocol Handler Version Numbers

Secret Server

> See full Secret Server Cloud Change Log for more details.

Table: Secret Server Cloud Distributed Engine Versions

Secret Server Version	DE Version	PH Version	Date Published	Required Upgrade	Notes
2025-01-23	8.4.45	6.0.3.33	2025-01-23		**PH**With this version, protocol handler improvements include resolving an issue where all metadata was displayed when more than 10 fields were added to a secret, ensuring proper metadata management. Updating the heartbeat on a secret now generates an audit entry, providing clear documentation of whether the heartbeat was enabled or disabled. Large SSH keys can now be imported on Linux servers, enhancing compatibility and functionality for users managing SSH keys. When a secret is erased, the TOTP key and backup codes are also erased, ensuring complete removal of sensitive information. A cooldown period of 1 hour has been introduced for the Quantum Lock password, improving user experience by reducing the frequency of password entry. An issue with the Secret

Secret Server Version	DE Version	PH Version	Date Published	Required Upgrade	Notes
					Dependency Failure default report not running has been fixed, ensuring accurate reporting and monitoring.
2024-08-03	8.4.33.0	6.0.3.28	2024-08-03		**PH:** With the previous version, protocol handler received changes to core internal functionality that prevents automatically updating to version 6.0.3.27 from a prior version. In environments with protocol handler automatic update enabled, the protocol handler will automatically update to version 6.0.3.26. To use the latest functionality and fixes of protocol handler, you must manually redeploy or install version 6.0.3.27 to end-user machines. Following that update, the automatic update will continue to work as before and upgrade to 6.0.3.28.

Secret Server Version	DE Version	PH Version	Date Published	Required Upgrade	Notes
2024-04-20	8.4.24.0	6.0.3.27	2024-04-20		**PH:** With this version, protocol handler received changes to core internal functionality that prevents automatically updating to version 6.0.3.27 from a prior version. In environments with protocol handler automatic update enabled, the protocol handler will automatically update to version 6.0.3.26. To use the latest functionality and fixes of protocol handler, you must manually redeploy or install version 6.0.3.27 to end-user machines. Following that update, the automatic update will continue to work as before.

Secret Server Version	DE Version	PH Version	Date Published	Required Upgrade	Notes
2024-02-10	8.4.21.0	6.0.3.27	2024-02-10	PH: Yes	**PH:** With this version, protocol handler received changes to core internal functionality that prevents automatically updating to version 6.0.3.27 from a prior version. In environments with protocol handler automatic update enabled, the protocol handler will automatically update to version 6.0.3.26. To use the latest functionality and fixes of protocol handler, you must manually redeploy or install version 6.0.3.27 to end-user machines. Following that update, the automatic update will continue to work as before.
2023-09-23	8.4.16.0	6.0.3.26	2023-09-23		**PH:** The legacy automatic update works up to this version. The next version will require a manual update. After that, the new automatic update will work. See the notes above.

Secret Server On-Premises

Table: Secret Server On-Premises Distributed Engine Versions

Secret Server Version	DE Version	PH Version	Date Published	Required DE/PH Upgrade	Notes
11.7.000016	8.4.32.0	6.0.3.28	2024-06-12	DE: Yes	**DE:** Replacement for 11.7.000015. We pulled down version 11.7.15 (on-premises) to resolve a problem with older versions of RabbitMQ (before 3.10) impacting DE version 8.4.31, which is shipped with SS 11.7.15. This issue does not impact Cloud.
11.7.000015	8.4.31.0	6.0.3.28	2024-05-22		**DE:** Pulled down to resolve a problem with older versions of RabbitMQ (before 3.10) impacting DE version 8.4.31, which is shipped with SS 11.7.15. This issue does not impact Cloud.
11.7.000002	8.4.24.0	6.0.3.27	2024-05-15		
11.7.000001	8.4.24.0	6.0.3.27	2024-04-13	PH: Yes	**PH:** With this version, protocol handler received changes to core internal functionality that prevents automatically updating to version 6.0.3.27 from a prior version. In environments with protocol handler automatic update enabled, the protocol handler will automatically update to version 6.0.3.26. To use the latest functionality and fixes of protocol handler, you must manually redeploy or install version 6.0.3.27 to end-user machines. Following that update, the automatic update will continue to work as before.

Secret Server Download Hashes

This topic only applies to **Secret Server On-Premises**.

Download hash codes or hashes are used to verify the integrity and authenticity of downloaded software. Hash codes are unique mathematical values that are generated based on the content of the software file and can be used to confirm that the downloaded file matches the original version and has not been tampered with. By comparing the hash code of the downloaded file with the hash code provided by the software developer, you can ensure that the software you downloaded is genuine and has not been corrupted during the download process.

11.7.000061

Calculated on 2025-01-31 18:04:03+00:00

Delinea Setup (11.7.000061):

- SHA1 = cd9bdb1288ca107eac29dfb1348045e4be30cc7f
- SHA256 = 6e5f4e1aaa07be82682011bc1ad016e3b56c45649b3ff3e03ce7444c2e5f2a18

Version_11_7_000061.zip:

- SHA1 = a5f786ce034894f030fa014388cbd47a6050feea
- SHA256 = 66a66d8cb71caa83d5ca3d809bb87b6ef0a0f52b2097738bedcbd7484536705e

11.7.000060

Calculated on 2025-01-06 20:37:37+00:00

Delinea Setup (11.7.000060):

- SHA1 = e59584200ba848a363dee240dc191ec09b234cdf
- SHA256 = 9306e9e8e64e1e29fe206457116676df3802ecf27c6aee91ce491f8c90b0a960

Version_11_7_000060.zip:

- SHA1 = 4b40859eacecb13af8f37f04594d8c1500a1f8b3
- SHA256 = c13b226a18b0a06d92e78f650289988f503e67827b04b3fae3432eee087bb024

11.7.000049

Calculated on 2024-11-21 21:05:03+00:00:

Delinea Setup (11.7.000049):

- SHA1 = 25d8c0b6de2aadbf8303071bec3a8d43d9b8f5c2
- SHA256 = 0b4890db0120a7976005f4c32b36d5804589d30539690ac490fd99c628b0a0ef

Version_11_7_000049.zip:

- SHA1 = 2974b07ca1ab469c465c330717399e1260830a75
- SHA256 = 840cfd275912e2b8490fbf60a6ccc3ac38d8f73ce205877755798d60cacba828

11.7.000031

Calculated on 2024-09-03 18:13:49+00:00

Delinea Setup (11.7.000031):

- SHA1 = 177b582e724953505ebcf956f4138e28c30e9012
- SHA256 = 2404b96cfa3835c2c4d9c644cac6e4d72b5eed1a6df855f544753756ec6676c0

Version_11_7_000031.zip:

- SHA1 = 277c7cbfce5d2faabbcc5e1bc770dcbab9b389ee
- SHA256 = 4e79f5f55d1938d9b558da19ddd0e72c756696a0dca4de737cc322e45470b09d

11.7.000016

Calculated on 2024-06-03 20:43:30+00:00

Delinea Setup (11.7.000016):

- SHA1 = fb3a018902dd62eb0f48d05ede4b556237a74bb7
- SHA256 = aa25f08639ce7568cfe0e3b1b85ffde084f9bc69e1f74ba4a7fed42ab3cb8c5a

Version_11_7_000016.zip:

- SHA1 = 55fbaf3c75434a115ca564342ca00769dcd64512
- SHA256 = 6ad35eb6cd4e7f43d9aa5fe6d6e04d811279b23a165cc6daa32c640334f99931

11.7.000015

Calculated on 2024-05-20 21:29:15+00:00

Delinea Setup (11.7.000015):

- SHA1 = 8ed1cf712afe8d921d308f585734c564d29ecfd7
- SHA256 = 6f997ec298d237eb8316fb2d85ee70b876abb0ac9d81923a6bc8998fe5942680

Version_11_7_000015.zip:

- SHA1 = 3b979f74a873dd9e95dce2c0bc8cf4dfaf525151
- SHA256 = a9a458d987036583f8db7019c6966b87075d9aff6f0e9a6d5bae1deb5ad1b656

11.7.000002

Calculated on 2024-05-09 20:25:39+00:00

Delinea Setup (11.7.000002):

- SHA1 = 1d5493dad8835c1a921cd22635e0ddda87c8cce6
- SHA256 = 3ca7ebe88dc0c4a095d6dbaab6b36d5922c8ed06d9033e6d40b45688974be7c8

Version_11_7_000002.zip:

- SHA1 = 29730ee8efd7144707c18a67741c32d851b398aa
- SHA256 = ebae89dc0994fe228d759a0aac1f448c4fb2af2a9198a784167e1229ae8e829d

11.7.000001

Calculated on 2024-04-13 02:41:13+00:00

Delinea Setup (11.7.000001):

- SHA1 = 4ecd4a2f8ccb6f903d0da8bc1657446b566d70e5
- SHA256 = fea3b79893ae57efb23db4049f92f0e312866f8c0aa7170eaf8cb4ada472addb

Version_11_7_000001.zip:

- SHA1 = 7523dc657cdddf1c94ba10da4fcc908bc68eae14
- SHA256 = 2dd27f06e7550a882a4a55a3a03b0ada48404c165afccd50a80d08322da32941

11.7.000000 GA

Calculated on 2024-03-26 18:13:30+00:00

Delinea Setup (11.7.000000):

- SHA1 = e453733d3ddeb467fa5bd8f9c0c8455467385137
- SHA256 = 98ed19e791eceda83d7f72812c7a1b642ef5c77d5c73ecfdbb21d46b70c0baf5

Version_11_7_000000.zip:

- SHA1 = 170b40eaf65a0c354fb10228f4ff099ab7a36f74
- SHA256 = 35453a1665c53d8c5f14edbdfdf97fdb733847f727eb0a1c4385c0b1d29b6658

11.7.000000 EA

The 11.7.000000 EA (Early Access) release number is officially 11.6.000043.

Calculated on 2024-03-01 16:21:11+00:00

Delinea Setup (11.6.000043):

- SHA1 = 1fd2202a54c0930a2cb35b634d96465d85b0f873
- SHA256 = 86fcb7981037f17a97d2735d8ab01768429d0e42602475630f6ddc1955a7fb08

Version_11_6_000043.zip:

- SHA1 = 2d850651fca8eadf4e2c903cd7af72d430dd7471
- SHA256 = 889dc210d3f94053465c9f463c98c469cfac1d5ecf622dac916f4f62b4

11.6.000025

Calculated on 2024-01-23 09:54:46-05:00

Delinea Setup (11.6.000025):

- SHA1 = 7368cd1fd99b2ce138d4307bb05681705a9fc395
- SHA256 = 9a7c16517f60ebf6787a83879681f0a2a59200f9158185aec0c9ed4f224bf556

Version_11_6_000025.zip:

- SHA1 = dba014aa2ed49d9e7965d6f92096d83aeb43bf11
- SHA256 = d49e927a1bde5b0b99a26bf004c1e4eab7912a4e64305c363af072a7d76fd37d

11.6.000004

Security Release

Calculated on 2023-12-01 13:31:29-05:00

Delinea Setup (11.6.000004):

- SHA1 = 40af6fa7f80e4020745d6d42399146a02b65fb69
- SHA256 = 9b539dddcaac51ca5d0cf56bddbcc110d5c81d176ad0e4db0e2d770293e11526

Version_11_6_000004.zip:

- SHA1 = 22ba4d14ec61a9c6d5e75636d87036c06ed1bcf2
- SHA256 = e2bec379218ebba11d0bfc95d7b56c7396771872c8d016204cc2ee71ce3f7c95

11.6.000003

Silent Release (October 2, 2023 or Later):

Calculated on 2023-09-29 16:44:43-04:00

Delinea Setup (11.6.000003):

- SHA1 = 6f591b3665d9f06dd524e6479c6cbea9f9b96195
- SHA256 = b1ed8bdd7808e83ce0b678cc4b516cf6ea575f9aaa74c3e711d10d1a9a477967

Version_11_6_000003.zip:

- SHA1 = c7b4a8a90af32091d5f1d8942a7437c5a98c3339
- SHA256 = efbb2c3eb0958ad1154a8944a341607bc945a5461c3408aa582ef9545f159324

Original Release (Before October 2, 2023):

Calculated on 2023-09-27 17:15:43-04:00

Delinea Setup (11.6.000003):

- SHA1 = 2db102b6082c74c665936f4b2c8b1616dc9ec0b3
- SHA256 = dfc4bc0a2f00d85e54db3d6da04f6e33fdcd634838dd053d36810e01cbf7c79f

Version_11_6_000003.zip:

- SHA1 = 65a8d71566f57ad02e90a48d844c77e29d1f6ab8
- SHA256 = 20f064774f4c4e8a50b9b5125277ed4d353ed15691075c01a4f9648dc4132708

11.6.000002

Calculated on 2023-09-22 16:38:43-04:00

Delinea Setup (11.6.000002):

- SHA1 = 29671d220b96f8315844ecd51cc6a5ac52e060ce
- SHA256 = 867c37621d48889372733cba9652539ad9a2e541ab8d8afbca3ccab66721f043

Version_11_6_000002.zip:

- SHA1 = 96df8277c0817299d8136c0ce0cd00697ad8c4a7
- SHA256 = 953455e98e3a875e1713deb7a90cea4c4aa6d9477b35b87cefde96167a3444b4

11.6.000000

Calculated on 2023-09-10 21:13:53-04:00

Delinea Setup (11.6.000000):

- SHA1 = dce9b9c2edb8003e074ff080be0ef8fcb0546983
- SHA256 = d38684eaff9f8e05e5d0df1a8ec1f2ae1707433d4b295d08b5ee41bc18f4a0b4

Version_11_6_000000.zip:

- SHA1 = 95f4edd8cf5df2bfbd9ac529ecf9ab1695735ef9
- SHA256 = 40abbc0f6ef6afd3f7b095d6bd97769b013a3d047cca2c8b5453080dc2f343a0

11.5.000002

Calculated on 2023-07-17 00:38:33-04:00

DelineaSetup.exe (11.5.000002):

- SHA1 = 5ec0e0d36e311e99d3caea6a7db058a3460d1509
- SHA256 = f04b3101a280a8168decfb56193eb10691fc4a30e4cab374e8b3ad73f9d28b8e

Version_11_5_000002.zip:

- SHA1 = 01644b1c4db374702e2bc14890a112a87f497162
- SHA256 = c7eb2871228aed268ca1d95da2ab610cb10f791f00533da1af955939ef2026cf

11.5.000001

DelineaSetup.exe (11.5.000001):

- SHA1 = 0a92d854096510a1df0e93bdcec61e4d7654e707
- SHA256 = 058731a8a55c13661e4e9062b19b29202f9dcbd8aeac30e550b94ae449acccfa

Version_11_5_000001.zip:

- SHA1 = 357aed1e30cb61968598511d7432d36ff891a6e7
- SHA256 = 2faa1d49ff4002da4bfa66a1a640aaadcc83ce535fc67c1e63d7ee9eeb867db4

11.5.000000

DelineaSetup.exe (11.5.000000):

- SHA1 = 33817b6b2d6aa3c0e32171a1756e084ee8631f9f
- SHA256 = bc75aab5bb7ec1e6335e8f743791d13324ded407c542f0cad5935f02eacd848c

Version_11_5_000000.zip:

- SHA1 = 9a1cf2ba64ed513283833cf2ededd0fea35bd9a9
- SHA256 = c2b75f8ffdb91a862ab6223caaffbaa4071460c20fffbedea294ace4f50daf28

11.4.000032

Retroactive security patch

Version_11_4_000032.zip:

- SHA1 = b8ea83c70b8b3e74956e28ab38cf2f99b2555914
- SHA256 = f80b3b36d20e8f001310adb40dd84c6c7f9f9e5a689b11e12d491ea63248c878

11.4.000031

DelineaSetup.exe (11.4.000031):

- SHA1 = a5cfb030398ded365ee7ea184f737d9a49d6c377
- SHA256 = efe2ce041bcc7d9864d8b396ceb015e4ac2b3f9cf5e8e5124e7ab18ba9238473

Version_11_4_000031.zip:

- SHA1 = 1cb41fcfd717692b13b21856d7b9055234dab1fe
- SHA256 = 06e0813ddc592a88054ca59e59c93006d2a2e22bec75cc0a4f55190ade77cf90

11.4.000030


Calculated on 2023-04-26 12:27:23-04:00

DelineaSetup.exe (11.4.000030):

- SHA1 = ae2c2441cf5fa0dd9b9d6c733906cc67dec4b245
- SHA256 = a97748c6315cb2d0540954f8e26d0baef0bbd16c26b2b5a2bcafcb8231bc0b84

Version_11_4_000030.zip:

- SHA1 = eb438fdf1339cbd9c1b6ffd86aea52027bfd6781
- SHA256 = b43073d69a60884b6419dabe04d059d1c6c317831cdd340aae17cbc712c1696f

11.4.000002 (GA)


Calculated on 2023-03-02 11:06:16-05:00

DelineaSetup.exe (11.4.000002):

- SHA1 = eac97741e09a79fc1cea04d4dce7a7c639ad6705
- SHA256 = edee604d844ad6ebbe33fe4b6cf0db1576bac67c33a0a65573be76ba275ac791

Version_11_4_000002.zip:

- SHA1 = 53295e58c75875d7e0e6ac3c687cf2e97ab4eca3
- SHA256 = ff4a3d7115d7fd39b86a6e58e268044adac2bb1a91a8495915831d5811c26332

11.4.000000 (EA)


Calculated on 2023-02-15 16:08:14-05:00

DelineaSetup.exe (11.4.000000):

- SHA1 = 9b905c8cb0dc891a1a15dc4ad0ac05b4de77fd57
- SHA256 = c2189de5e70baaa00c7998e58036a49812ebdf857d17d719909352ccad7fc0fb

Version_11_4_000000.zip:

- SHA1 = 0f3f369995434c942a56190b270954f0093e9c84
- SHA256 = dac53799fddd6063b066bec58298de831e28864a4fb668b3c78b61aadc714ad9

Downloading Secret Server

This topic only applies to **Secret Server On-Premises**.

To download Secret Server, navigate to Secret Server Download to get the latest .zip file. See the other topics in this section for installation or upgrade.

IIS and Secret Server

This topic only applies to **Secret Server On-Premises**.

Secret Server uses Internet Information Services (IIS) to run its web application, providing a secure and scalable platform for managing privileged credentials. Here are some key ways in which Secret Server uses IIS:

- Web Hosting: Secret Server is hosted on IIS, which serves as the web server to deliver the application to users. IIS handles HTTP/HTTPS requests, ensuring that the web application is accessible over the network.
- Authentication: Secret Server uses IIS to manage various authentication methods, including Integrated Windows Authentication (IWA). This allows users to authenticate using their Active Directory credentials without needing to re-enter them, enhancing security and user convenience.
- SSL/TLS Encryption: IIS is configured to use SSL/TLS to encrypt data transmitted between the client and the server. This ensures that sensitive information, such as passwords and secrets, is protected from eavesdropping and man-in-the-middle attacks.
- Application Pool Management: Secret Server runs in its own application pool within IIS. This isolates the application, ensuring that it has dedicated resources and can be managed independently of other applications on the server. The application pool can be configured to always run, preventing it from stopping due to inactivity.
- Logging and Monitoring: IIS provides logging capabilities that Secret Server utilizes to track access and activity. This includes logging user actions, system events, and errors, which are crucial for auditing and troubleshooting.
- Security Hardening: Secret Server employs various IIS security features to harden the application against attacks. This includes disabling unnecessary HTTP headers, configuring secure cipher suites, and ensuring that the application pool runs with the least privilege necessary.
- Session Management: IIS manages user sessions, ensuring that authenticated sessions are maintained securely. This includes handling session timeouts and ensuring that sessions are properly terminated when users log out.
- Load Balancing and Clustering: For high availability and scalability, Secret Server can be deployed in a clustered environment using IIS. This allows multiple instances of Secret Server to run simultaneously, distributing the load and providing redundancy in case of server failure.

Manual IIS Installation

This topic only applies to **Secret Server On-Premises**.

See "Advanced (Manual) Installation" on page 65 for more information.

IIS is an internal part of the Windows operating system, and only needs to be enabled. If IIS is not found, the Delinea Installer will install it for you. If you would prefer to install IIS manually, please refer to the instructions listed below for example steps in the Windows Server 2016 Operating System. For the most up-to-date setup instructions, see Microsoft's Technical Documentation. Navigate to **Docs > Internet Information Services > Install**.

Roles and Features

Delinea products recommend the following roles and features to be installed on the Secret Server IIS Server for maximum security and functionality options:

Roles

- Web Server (IIS)
- Web Server (IIS)\Web Server
- Web Server (IIS)\Web Server\Common HTTP Features
  - Default Document
  - Directory Browsing
  - HTTP Errors
  - Static Content
  - HTTP Redirection
- Web Server (IIS)\Web Server\Health and Diagnostics
  - HTTP Logging
- Web Server (IIS)\Web Server\Performance
  - Static Content Compression
  - Dynamic Content Compression
- Web Server (IIS)\Web Server\Security
  - Request Filtering
  - Windows Authentication
- Web Server (IIS)\Web Server\Application Development
  - .NET Extensibility 4.6
  - Application initialization
  - ASP.NET 4.6

  - ISAPI Extensions
  - ISAPI Filters
- Web Server (IIS)\Web Server\Management Tools
  - IIS Management Console
  - IIS Management Scripts and Tools

Features

- .NET Framework 4.x Features
  - .Net Framework 4.x
  - ASP.NET 4.x
  - WCF Services:
    - HTTP Activation
    - TCP Activation
    - TCP Port Sharing
- Windows PowerShell
  - Windows PowerShell 5.1
- Windows Process Activation Service
  - Process Model
  - Configuring APIs

Step One: Windows Server 2012-2019 IIS Installation

To install Internet Information Services (IIS) Manager on Windows Server 2016, you will need to give your server the Web Server (IIS) role using the following procedure:

> If this is *not* the first time you have run the wizard (that is, when first installing IIS), the Web Server Role (IIS) and Role Services windows will not appear, and the wizard order changes a bit. Instead, role services are selectable in the Server Roles window.

1. Click the **Server Manager** button on your server. The Server Manager Dashboard appears.
2. Click the **Add Roles and Features** button. The Add Roles and Features Wizard on the Before You Begin window appears.
3. Click the **Next** button. The Select Installation Type window appears.
4. Click to select **Role-based or feature-based installation** selection button.
5. Click the **Next** button. The Select Destination Server window appears.
6. Ensure the **Select a Server from the Server Pool** selection button is selected.
7. In the **Server Pool** section, click to select your server.

8. Click the **Next** button. The Select Server Roles window appears.
9. Click to select the **Web Server (IIS)** check box.
10. Click the **Next** button. The Select Features window appears.
11. In the **Features** list, Click to select the following checkboxes (If necessary, click the **Add Features** button when prompted):
    - .NET Framework 4.x Features > WCF Services > **HTTP Activation**
    - .NET Framework 4.x Features > WCF Services > **TCP Activation**
12. Click the **Next** button. The Web Server Role (IIS) window appears.
13. Click the **Next** button. The Select Role Services Window appears.
14. In the **Roles** list, click to select the following check boxes:

    Leave all the auto-selected check boxes as is.

    - Web Server (IIS) > Web Server > Common HTTP Features > **HTTP Redirection**
    - Web Server (IIS) > Web Server > Performance > **Dynamic Content Compression**
    - Web Server (IIS) > Web Server > Security > **Windows Authentication**
15. Click the **Next** button. The Confirmation window appears
16. Confirm your installation details.
17. Click the **Install** button. Wait for the installation to complete. The Results window appears.
18. Click the **Close** button. An IIS tile should now appear on your server.

We recommend you run Windows Update to install the latest security patches for IIS once you have IIS installed.

Step Two: Configure the IIS Website

Follow these steps to configure a website in IIS for Secret Server:

1. Extract the Secret Server files into `C:\inetpub\wwwroot\SecretServer`or your location of choice. If you rename`SecretServer`, do not use more than 20 characters.
2. Open Internet Information Server (IIS) Manager: On the taskbar, click **Server Manager > Tools > Internet Information Services (IIS) Manager**.
3. In the Connections pane, expand the server name.
4. Click on the **Application Pools** node. The Application Pools window appears.
5. Click the **Add Application Pool** link. The Add Application Pool dialog box appears.
6. Type `SecretServer` in the **Name** text box.
7. Click to select **4.x** in the **.NET Framework Version** dropdown list.
8. Click to select **Integrated** in the **Managed Pipeline Mode** dropdown list.

9. Click the **OK** button to save the new application pool. The dialog box closes.
10. (optional) Customize the Windows account Secret Server runs as:
    a. Right click the new application pool and select **Advance Settings…**
    b. Click the **Identity** setting in the **Process Model** section to select the desired account. Using this, you can, for example, set Secret Server to use IWA to connect to SQL.
11. Expand the **Sites** node on the **Connections** tree.
12. Click on the Default Web Site node.
13. In the **Actions** pane, click **Bindings** to set your desired website. The Edit Bindings dialog box appears.
14. Edit or add bindings as desired. We recommend using HTTPS with a real SSL certificate.
15. Click the **Close** button.
16. In the **Connections** tree, expand the **Default Website** node.
17. **Either**, If you see the default folder, **SecretServer**, which you created earlier:
    a. Right click the **SecretServer** folder and select **Convert to Application**. The Add Application dialog box appears.
    b. Click the **Select…** button to choose the pool you created earlier for Secret Server.

    Or, If you used a custom location instead:

    a. right click the Default Website. The Add Application dialog box appears.
    b. Type `SecretServer` in the **Alias** text box.
    c. Click **Select…** and pick the app pool created for Secret Server.
    d. Type the path where you extracted the Secret Server files in the **Physical Path** text box.
18. Click the **OK** button.

Step Three: Ensure IIS Does Not Stop the Worker Process

When using IIS version 7.0 and above, by default, the worker process terminates after an inactive period. If Secret Server is in its own application pool, that application pool will stop after a period of no requests. To ensure this does not happen, perform the following procedure. Additionally, by default, IIS launches a worker process when the first request for the Web application is received, so if the Secret Server application takes a long time to start, issues can result. Thus, we recommend launching the Secret Server application pool worker process as soon as IIS starts by setting the start mode to "AlwaysRunning."

Procedure:

1. Open **Internet Information Server (IIS) Manager**:
    - If you are using Windows Server 2012 or Windows Server 2012 R2: On the taskbar, click **Server Manager > Tools > Internet Information Services (IIS) Manager**.
    - If you are using Windows Server 2008 or Windows Server 2008 R2: On the taskbar, click **Start > Administrative Tools > Internet Information Services (IIS) Manager**.
2. In the **Connections** pane, expand the server name.

3. Click **Application Pools**.
4. Determine which application pool Secret Server is running as:
   a. Expand **Sites** at the left.
   b. Find the website Secret Server is running on.
   c. Click on the Secret Server website or virtual directory (if it is running on one).
   d. Click **Basic Settings** on the right panel. This indicates Secret Server's application pool.
5. Right-click the application pool and select **Advanced Settings...** The Advance Settings dialog appears.
6. In the **General** section, set **Start Mode** to **AlwaysRunning**.
7. In the **Process Model** section, set **Idle Time-out (minutes)** to **0**.
8. In the **Recycling** section, set **Regular Time Interval (minutes)** to **0**.
9. In the **Recycling** section, click the **>** next to **Specific Times** to ensure there are no times set. If there are, click the **...** to clear them.
10. Leave IIS Manager open—we will return to it below.

Step Four: Ensure the User Profile Always Loads

As of version 10.2, Secret Server requires its application pool "Load User Profile" setting enabled. Otherwise, Secret Server reports a critical alert to system admins.

> Even without the setting enabled, Secret Server loads to give access to secrets but many internal operations may malfunction, so we recommend resolving this issue as soon as possible.

Procedure:

1. Right-click the Secret Server application pool in IIS Manager and select **Advanced Settings...** The Advance Settings dialog appears.
2. Go to the **Process Model** section in the **Advanced Settings** dialog.
3. Set **Load User Profile** to **True**.
4. Preform an `iisreset` on the server (in an administrator command prompt).

Running the IIS Application Pool As a Service Account

> This topic only applies to **Secret Server On-Premises**.

Overview

A domain service account should be used for the Secret Server application pool that can both access the Delinea product's SQL database and run the IIS Application Pool(s) dedicated to your Delinea product.

> A service account will usually need access to both the IIS resources and the database, but there may be cases when this may not be required.

> The service account created in this KB should **not** be the same account that is created during the installation of SQL and used to manage SQL as a whole.

To set up this service account correctly you will need to:

1. Create a service account in Active Directory that will be dedicated to your Delinea product (domain).
2. Granting the service account access to the SQL Server database.
3. Assign the service account as the identity of the application pool or pools in IIS.
4. Grant folder permissions for the service account on two folders.
5. Configure User Rights Assignment to the service account.

Procedure

> You must have IIS installed on your Web server before completing these steps.

Task 1: Creating a Domain Service Account

1. Create a local or domain user account (or identify one to use).
2. Open IIS (**Search > inetmgr**) on your Web server.
3. Open the **Active Directory Users and Computers** link from **Administrative Tools**.
4. Click to open the directory where you want to assign this account, such as testlab.com.
5. Select **Service Accounts**.
6. Right click and select **New > User**. The "New Object - User" wizard dialog box appears.
7. Type a name and logon name for the service account.
8. Click the **Next** button. The wizard advances to the next dialog box (same name).
9. Type a password in the Password and Confirm Password text boxes.
10. If necessary, click to deselect the **User must change password at next login** check box.
11. Click to select the **Password never expires** check box. Failing to do this could lock the account out of Secret Server.
12. " Check **Password never expires** or the account could lock you out of Secret Server.
13. Click **Next** button.
14. Click the **Finish** button. You can now give the account access to the database server and the application server.

Task 2: Granting Access to the SQL Database

> You must have SQL installed on your database server before completing these steps.

Grant access:

1. Open the SQL Management Studio on your database server.
2. Connect to your Delinea product's SQL database using an administrator account.
3. Click to select the Security folder in the Object Explorer.
4. Right-click the same folder and select **New > Login...**. A log on dialog box appears.
5. Ensure the **Windows Authentication** radio button is selected.
6. Click the **Search...** button. The "Select User, Service Account, or Group" dialog box appears.
7. Ensure that your domain or AD server appears in the **From this location** text box. If not, click the **Locations...** button and select it.
8. Type the login name you created for your Delinea service account, such as svc_thycotic, in the **Enter the object name to select** text box.
9. Click the **Check Names** button.
10. Click to select the correct account.
11. Click the **OK** button. The dialog box closes, returning you to the Login - New dialog box.
12. **Either**, if you have already created the database for your Delinea product:
    a. Click **User Mapping** in the **Select a page** list box.
    b. Click to select the check box for the database in the **Users mapped to this Login** list.
    c. Click to select the **db_owner** check box in the **Database role membership...** list.
13. **Or**, if you have not yet created the database:
    a. Click **Server Roles** in the **Select a page** list box.
    b. Click to select the **db_creator** check box.
14. Click the **OK** button.

Task 3: Assigning the Identity of Application Pools

1. Click the **Applications** node under the server name in the **Connections** tree.
2. Right-click the node and select **Advanced Settings...** The Advance Settings dialog box appears.
3. Click the **...** button for the **Identity** entry in the **Process Model** section. The Application Pool Identity dialog box appears.
4. Click to select the **Custom Account** selection button.
5. Click the **Set...** button. The Set Credentials dialog box appears.
6. Type your service account's name, such as test and password.
7. Click the **OK** button. The dialog box closes.
8. Open the command console as an Admin.
9. Change the directory to your .NET framework installation directory using the "cd" command, for example, `C:\Windows\Microsoft.NET\Framework\v4.0.30319`.

10. Type `.\aspnet_regiis -ga <domain name>\<username>`, replacing `<domain name>` and `<username>` with your information. For local accounts omit the domain name parameter.

Task 4: Granting Folder Permissions

> You must have the Delinea product application files installed (on your Web server) before completing this section.

Following the steps below, you give the service account "Modify" access to **two** folders:

- `C:\Windows\TEMP`
- The folder where your Delinea product's application files are located, such as `C:\inetpub\wwwroot\SecretServer`

Procedure (for each folder):

1. In a file manager, navigate to the Secret Server application folder.
2. Right-click the folder and select **Properties**. The Properties dialog box appears.
3. Click the **Security** tab.
4. Click the **Advanced** button.
5. Click the **Add** button. A permissions panel appears.
6. Click the **Select a Principal** link. The "Select User, Computer, Service Account, or Group" dialog box appears.
7. Ensure that your domain or AD server appears in the **From this location** text box. If not, click the **Locations...** button and select it.
8. Type the login name you created for your Delinea service account, such as svc_thycotic, in the **Enter the object name to select** text box.
9. Click the **Check Names** button.
10. Click to select the correct account.
11. Click the **OK** button. The dialog box closes, returning you to the permissions panel.
12. Click to select the **Modify** check box in the **Basic Permissions** section. Your service account should have the **Modify**, **Read & Execute**, **List folder contents**, **Read**, and **Write** permissions selected for this folder
13. Click the **OK** button.
14. Click the **Apply** button.

> If a Windows Security pop-up appears, click the **Yes** button. The service account will now be able to access this folder.

> The application folder only needs "Write" and "Modify" permissions during the installation or during an upgrade. You can remove these once the installation process is complete.

> In order to encrypt the `encryption.config` file, the App Pool identity account needs permissions on the file.

Task 5: Configuring User Rights

The following settings are required for DelineaSecret Server to function:

- "Log on as a batch job"
- "Impersonate a client after authentication"

You can adjust these settings either at the **Domain** level using group policy or locally on your IIS Web server using the Local Security Policy Console. See User Rights Assignment to learn more.

Option 1: Setting User Rights Assignment on the Domain

> This is an example of how to create a Group Policy Object (GPO), we recommend consulting with your organizational group policy administrator to create this policy.

> This overwrites any configuration in the local security policy. The local security policy is a safer option if you are not sure about usage across your domain.

1. Open the Group Policy Management Console.
2. Right-click the desired GPO folder (under the domain node) in the **Group Policy Management** Tree, and select **New**. The New GPO dialog box appears.
3. Type the name, such as " Delinea User Rights Assignment," in the **Name** text box.
4. Click the **OK** button. The dialog box closes.
5. Right-click the GPO you just created and select **Edit**. The Group Policy Object Editor appears.
6. On the **Computer Configuration** node, click to expand **Policies > Windows Settings > Security Settings > Local Policies**.
7. Click to select the **User Rights Assignment **folder.
8. Repeat the following procedure for the "Log on as a batch job" and "Impersonate a client after authentication" permissions (for this instruction we show the former):
   a. In the list on the right, right-click **Log on as a batch job** and select **Properties**. The "Log on as a batch job Properties" dialog box appears.
   b. Ensure that the **Define these policy settings** check box is checked.
   c. Click the **Add User or Group** button. A dialog box appears.
   d. Add your Delinea service account.
   e. Click the **OK** button. The dialog box closes. The new policy appears in the list.
   f. Click the **Apply** button.

9. Link your new GPO to the OU where your Delinea product machine accounts exist, that is, the Web and database servers.

Option 2: Setting User Rights Assignment Locally

1. On the Web server hosting IIS and your Delinea Application files, open the "Local Security Policy Console" as an administrator (Run as administrator).
2. On the Local Policies node, click to expand **Local Policies > User Rights Assignment**.
3. Click to select the **User Rights Assignment** folder.
4. Repeat the following procedure for the "Log on as a batch job" and "Impersonate a client after authentication" permissions (for this instruction we show the former):
   a. Right-click on **Log on as a batch job** in the list on the right and select **Properties > Add User or Group**.
   b. Click to select your Delinea service account.
   c. Click the **OK** button.

If you get a "Service Unavailable" error after applying "Log on as a batch job" permissions, try updating your group policy settings: Open the **Command Console**, type in `gpupdate /force`**, and restart the **Windows Process Activation Service**.

Installation

This topic only applies to **Secret Server On-Premises**.

Please check the "Prerequisites" on page 85 and then select either our "Basic (Automatic) Installation" on page 68 or "Advanced (Manual) Installation" below.

Advanced (Manual) Installation

This topic only applies to **Secret Server On-Premises**.

See "Manual IIS Installation" on page 56 for more information.

Procedure

For the highest scalability and reliability, Delinea recommends using RabbitMQ. MemoryMQ is an easier but less capable alternative and can be used for trials and proof of concepts but **should not** be used for production environments. Two exceptions are very small deployments and customers that do not use open-source software for compliance reasons.

Step 1: Downloading the Secret Server Application Files

> Ensure you have the IIS, .NET Framework, and SQL Server prerequisites installed before following the steps below.

Go to the download page to get a .zip file that contains both Secret ServerandPrivilege Manager files in the manual installation section. Use this .zip file for the instructions below.

Step 2: Creating Folders and Extracting Contents

1. Extract the contents of the .zip file downloaded above (Right-click, **Extract All...**). The original file is named with the latest version number for Secret Server.
2. Extracting this file reveals a `nugetCache` folder, as well as another zipped folder named `ss\_update`. For a Secret Server-only install, you will not need the contents of the `nugetCache` folder.
3. Create a folder called `SecretServer` in the location `C:\inetpub\wwwroot\`.
4. Extract the contents of the `ss\_update.zip` file (Right-click, **Extract All...**) to `C:\\inetpub\\wwwroot\\SecretServer`.

Step 3: Configuring IIS

Open Internet Information Services (IIS) Manager* and create a new application pool:

> Our IIS installation sets the .NET trust level to "Full (internal), which may affect other applications on the server.

1. Right-click **Application Pools** and select **Add Application Pool...**
2. Type a name (for example, SecretServerAppPool).
3. Ensure that the highest .NET CLR version is selected.
4. Ensure the Managed pipeline mode is set to **Integrated**.
5. Click the **OK** button.

> The Secret Server installer sets the application pool to default to the system Network Service account. If you selected Windows Authentication Mode during the SQL Installation process, see "Running the IIS Application Pool As a Service Account" on page 60. To use Windows Authentication you must use an Active Directory service account to run the application pool in IIS. We recommend this as a security best practice.

6. See "Changing IIS to Not Stop Worker Process in IIS 7.0 and Later" on page 238 to set the Idle Timeout and Regular Timeout settings to 0 for the application pool in IIS.
7. Install Secret Server as either a virtual directory (4a) or as a website (4b):

Step 4a: Installing Secret Server as a Virtual Directory

1. Right-click **Default Web Site** and select **Add Virtual Directory...**
2. Select an alias for your Secret Server. The alias is appended to the website, and it is best to name it the name of your earlier unzipped folder. For example, SecretServer becomes `https://myserver/SecretServer`.
3. Select the physical directory for where you unzipped Secret Server, for example, `C:\inetpub\wwwroot\SecretServer`. Do not replace`SecretServer`with anything longer than 20 characters.
4. Click the **OK** button.
5. In the tree, right-click the new virtual directory and select **Convert to Application**.
6. Set the **Application Pool** to the same one you created in the Manual Installation section, for instance, SecretServerAppPool. Secret Server is now ready for installation. Skip to Step 5.

Step 4b: Installing Secret Server as a Website

1. In IIS, right-click **Sites** and select **Add Website...**
2. Type a site name.
3. Click **Select...** and choose the application pool you created in the Manual Installation section.
4. Click the **OK** button.
5. Click the **...** button beside the **Physical path** field and select the directory containing the unzipped Secret Server files, for example `C:\inetpub\wwwroot\SecretServer`.
6. Click the **OK** button.
7. Click the **OK** button at the bottom of the **Add Website** window to save your settings. Secret Server is now ready for installation.

Step 5: Completing Secret Server Installation from the Website

Your Secret Server advanced installation is now ready to complete:

1. "Installing and Configuring SQL Server" on page 99.
2. Open a browser and navigate to where your Secret Server is located, such as `http://localhost/secretserver`. You should arrive at a page that says "Secret Server (Not Installed or Unable to Access the Database)."
3. Click the **Install Secret Server** button.
4. On the **SQL Server Location** page, specify the server name of your SQL Database Server, `<DatabaseMachineName>\InstanceName` and then the database name that you created in SQL for Secret Server.
5. If you are using Windows authentication mode to access SQL (recommended), ensure the correct service account is listed.
6. If you selected mixed mode during the SQL install, select **SQL Server Authentication** and enter the SQL username and password you created for the SQL account. For information about adding a SQL Server user, see the "SQL Server 2016 Standard Edition Installation" on page 111.

7. Click the **Install Secret Server** button. Secret Server verifies it is able to successfully create the Secret Server database. If an error occurs no database changes will be made.

   > Secret Server attempts to download and install the latest version from the Internet. If you do not have an active Internet connection on your Web server, Secret Server will continue to install the version from your downloaded application files.

8. The install may take a few minutes to complete. Once successful, click the **Return to Home** button.
9. Create a username and password for the administrator account for Secret Server and store these credentials in a safe location.
10. Click the **Create User** button and log on after entering the username and password.
11. Once logged on Secret Server, you are prompted with the Getting Started wizard. The wizard guides you through adding your Licenses, setting up an email server, and creating your first group.

    > If you skipped the wizard and would like to return, go to **HELP > Getting Started** from the top menu.

Secret Server is now installed. See our "Getting Started Tutorial Overview" on page 78 or contact Delinea Support about training.

Troubleshooting Notes

- If the database name you provide does not yet exist in the specified instance of SQL Server, Secret Server attempts to create the database using the SQL or Windows account you have specified. For that account to create a database, it needs to have the dbcreator server role in SQL Server. Secret Server
- If using Windows authentication mode (recommended) you need to use a service account to run SS's application pools with appropriate permissions. See "Running the IIS Application Pool As a Service Account" on page 60.

Basic (Automatic) Installation

> This topic only applies to **Secret Server On-Premises**.

Introduction

This is the installation guide for Windows Server 2012-2022 and Windows 10. For other operating system installation guides, please see "Technical Support" on page 72.

Secret Server Is an ASP.NET Website

Secret Server is installed as an ASP.NET website. The setup.exe file sets up the website with the correct permissions and creates the settings in IIS.

SQL Server Is Usually Required

Secret Server requires an instance of SQL Server for the database backend and is installed by the setup.exe file, if missing. The SQL Server database will require a SQL account with *db_owner* permission to complete the

installation.

> Delinea does not support using SQL Express in a production environment due to size and performance limitations.

Administrative Access

Throughout the installation, you will be required to be an administrator to perform most of these actions. Please ensure that you are logged onto your system with a Windows account that has administrative rights.

Review the Prerequisites

> Except for the operating system, the following prerequisites are installed automatically by our installer. If you already have some of them installed or wish to install them yourself, the installer will skip over them.

Please review the full list of system requirements and recommendations on the "System Requirements for Secret Server" on page 86 page.

Additional Recommendations

We suggest you:

- Use an SSL certificate for Secret Server.
- Run Microsoft Update on your server to make sure all components are up to date.
- For the highest scalability and reliability, Delinea recommends using RabbitMQ. MemoryMQ is an easier but less capable alternative and can be used for trials and proof of concepts but **should not** be used for production environments. Two exceptions are very small deployments and customers that do not use open-source software for compliance reasons.

Procedure

Step 1: Downloading the Latest Version of Secret Server

The latest version of Secret Server is available for downloadat . A setup.exe file is downloaded to your machine. We recommend running setup.exe as an administrator.

Step 2: Running the Installer

Welcome Page

The first installer page you are presented is the Welcome Page. The installer should detect whether the machine has Secret Server orPrivilege Manager for Windows and will declare which of those products it will install.

Database Page

The Database page allows you to choose to install SQL Express or connect to an existing SQL Server. If you select SQL Express, the installer requires Internet access to download the installation for SQL Server Express.

If Internet access is not available, a link to download SQL Server Express is presented. You are expected to install SQL Server Express and then restart the installer.

If Internet access is available, SQL Server Express is installed.

Prerequisites Page

The Prerequisites page ensures everything that is required to install Secret Server is setup correctly. Everything on this page *can* be installed outside of the installer. If not, the installer installs and configures them for you. This page is primarily for third party server configuration. If there are issues, please refer to support for the specific non-Delinea vendors.

Database Connection Page

The Database Connection page contains the connection information that Secret Server(and Privilege Manager) uses. You must click the**Test Connection** button and have a successful result before installation can continue.

Create User Page

The Create User page is where you enter the information for the initial Secret Server user.

Email Server Page

Enter connection information for the email server on this page. This is also optional and you can skip it and set it up later in Secret Server. This page will configures email for both Secret Server andPrivilege Manager for Windows.

Review Page

Review the, mostly default, settings on the Review page, and change them is needed. Some of the settings are validated before the install can begin.

Install Page

The Install page shows the status from log files as both Secret Server andPrivilege Manager are installed.

Step 3: Reviewing the Log Files (Optional)

After the applications are installed, the installer opens a Web browser to the Secret Server log on page. At this point, everything is installed to start using both Secret and Privilege Manager. If the installation failed or you wish you view the logs from the installation, click the **View Log Files** button.

Step 4: Opening Secret Server

If the setup.exe did not automatically open a browser, navigate to where Secret Server is located, for example: `http://localhost/secretserver`.

Step 5: Learning Secret Server

See our "Getting Started Tutorial Overview" on page 78 or contact Delinea Support about training.

Installing Secret Server via the Command Line


This topic only applies to **Secret Server On-Premises**.

Overview

ThycoticSetup.exe accepts command line arguments for a silent or automated installation. This topic discusses how to do that.

Basic usage:

```
ThycoticSetup.exe -q -s PARAMETER=<value> PARAMETER2=<value> (/nodetect) (/l <log file path>)
```

Important considerations:

- Always pass `-q -s` to `ThycoticSetup.exe` and then pass in your parameters or switches.
- There are two stages to the installer. The first (optional) stage is to install the prerequisites such as IIS and .NET 4.8. Then in a second stage, once all required prerequisites are present, you can install Secret Server or Privilege Manager (PM).
- The installer UI performs additional validation steps, such as testing the database connection information, that a silent CLI one does not. Thus, this install can fail if you provide incorrect settings.
- The installer checks to see if Secret Server and PM are already installed by default. It will install them if either is not found. If you would like to specify which applications to install, you must use the `/nodetect` switch to avoid the automatic detection, so the `InstallSecretServer` and `InstallPrivilegeManager` settings are respected.
- Due to how MSI installers work, if you need to pass in parameters that contain spaces, use the special `CMDLINE` parameter, using extra double quotes to delineate each parameter. For instance:

  ```
  ThycoticSetup.exe -q -s PARAM1=some_string PARAM2=1234 CMDLINE=" PARAM3=""Something with a space"" PARAM 4=""Another value with spaces"" "
  ```

  Note:

  - You can mix and match regular parameters (numerical values, or strings without spaces) with `CMDLINE`.
  - Any parameters sent inside of `CMDLINE` are treated as strings. Numerical parameters inside of `CMDLINE` are ignored.
- Be aware of how you call `ThycoticSetup.exe` and what special characters need escaping in your shell. This includes passwords containing symbols. What is required is shell dependent. For example, running the installer from PowerShell (or a .ps1 script) rather than an older command prompt (or a .bat file) would require escaping a different set of special symbols.
- We recommend using the `/l <logfile>` option to create a log file, which you can use to verify your parameters are correctly passed to the installer. This is especially useful when using `CMDLINE` for parameters with spaces, which is prone to mistakes.

  
  For security, parameters involving passwords are not logged.

Install Prerequisites

As of Secret Server 10.11, you can silently install all required prerequisites. These are the same prerequisites the "Fix Issues" button in the installer UI fixes. The important difference is missing prerequisites are not auto detected–you must tell the installer which ones you want installed. Older versions will not do a silent command line installation unless these necessary prerequisites are already installed.

Parameters

Table: ThycoticSetup.exe Parameters

Parameter	Value	Required (if not present)	Purpose
`InstallPreReqs`	Boolean	Yes	Triggers the prerequisites installation. Enable this to install prerequisites with no application. This overrides `InstallSecretServer=1` and `InstallPrivilegeManager=1`.
`PRE_REQS_TO_INSTALL`	Comma separated list (see below)	Yes	Specifies which prerequisites to install.

Prerequisites

Table: Prerequisite (`PRE_REQS_TO_INSTALL`) Values

Prerequisite	Required (if not present)	Purpose
`CONFIGURE_FIPS`	No	Ensures the AES 256 and 128 ciphers are enabled in Windows.
`ENABLE_FIPS`	No	Enables FIPS mode in Windows. Generally, not needed, unless required by your environment.
`INSTALL_HTTPS_BINDING`	No	Enables HTTPS binding in IIS for the default website. Tries to pick an existing valid SSL certificate and creates a self-signed certificate if necessary for temporary use. Always use HTTPS with a valid certificate in production environments.
`INSTALL_IIS`	Yes	Installs the Web Server (IIS) Windows Role.
`INSTALL_IIS_COMPS`	Yes	Installs various required IIS features.

Prerequisite	Required (if not present)	Purpose
INSTALL_NET_WCF	Yes	Installs the WCF HTTP and TCP activation features.
INSTALL_NetFx48	Yes	Installs .NET 4.8. This **requires a reboot.**

Single-Line Example

```
ThycoticSetup.exe -q -s InstallPreReqs=1 PRE_REQS_TO_INSTALL=INSTALL_IIS,INSTALL_IIS_
COMPS,INSTALL_NET_WCF,INSTALL_HTTPS_BINDING,INSTALL_NetFx48 /l C:\temp\install-prereqs.log
```

Installing Applications

Secret Server or Privilege Manager can be installed and pre-configured using these parameters. If the required prerequisites are not already present, the installer exits. They can both be installed at the same time but will then share the same database and email settings.

If you need more control over configuring the website, you can create a site and configure it in advance (that is, using IIS's `AppCmd.exe`), and then pass the preconfigured website name as `SecretServerSiteName` or `PrivilegeManagerSiteName`.

Secret Server Parameters

Table: Secret Server Parameters

Parameter	Value	Default	Notes
CreateWebSite	Boolean	0	Required if the SecretServerSiteName website does not exist.
InstallSecretServer	Boolean	1	Whether or not to install Secret Server. Must also use the /nodetect switch to avoid this being set to 1 if not already installed.
SecretServerApplicationName	String	SecretServer	Used for the application pool name as well as the website application or subfolder.

Parameter	Value	Default	Notes
SecretServerAppPassword	String	Optional	Only required if you are configuring SecretServerAppUserName.
SecretServerAppUserName	String	ApplicationPoolIdentity	What identity to run the app pool as. The user must already exist.
SecretServerConfigLogFile	String	<logname>_SS_Configuration.log	Optional. The base <logname> is specified with the /l option.
SecretServerDestinationFolderPath	String	C:\inetpub\wwwroot\SecretServer	If you would like to use a directory containing spaces, see note above about how to use the CMDLINE parameter to pass it in. Do not replace SecretServer with anything longer than 20 characters.
SecretServerSiteHttpsPort	Integer	Optional	HTTPS port. Always use HTTPS in production. If using the default website, this port is an additional HTTP binding, along with the default. If you use this option to bind HTTPS on another port, configure the HTTPS binding yourself and choose a certificate after the installer completes. INSTALL_HTTPS_BINDING only configures the certificate on the normal 443 binding.

Parameter	Value	Default	Notes
SecretServerSiteName	Integer	Default website	Used by both Secret Server and PM. Must already exist, unless you also use CreateWebSite=1 (see the Website Parameters section below).
SecretServerSitePort	Integer	Optional	HTTP port. If using the default website, this port HTTP binding is in addition to any defaults.
SecretServerUserDisplayName	String	None	The display name for SecretServerUserName.
SecretServerUserEmail	String	None	The email address for SecretServerUserName.
SecretServerUserName	String	None	The initial Secret Server administrator user. If not set, once Secret Server is installed, the first person to visit the website will be able to pick the details on the "Create Initial Administrator" page.
SecretServerUserPassword	String	None	The password for SecretServerUserName.

Privilege Manager Parameters

Table: Privilege Manager Parameters

Parameter	Value	Default	Notes
CreateWebSite	Boolean	0	Required if the PrivilegeManagerSiteName website does not exist.

Parameter	Value	Default	Notes
`InstallPrivilegeManager`	Boolean	1	Whether or not to install PM. Must also use the `/nodetect` switch to avoid this being set to 1 if not already installed.
`PrivilegeManagerApplicationName`	String	`TMS`, `TMSAgent`, `TMSWorker`	Used as the base name of the PM application pools (Regular, Agent, and Worker), plus the website application or subfolder.
`PrivilegeManagerAppPassword`	String	None	Only required if you are configuring `PrivilegeManagerAppUserName`.
`PrivilegeManagerAppUserName`	String	None	What identity to run the app pool as. User must already exist.
`PrivilegeManagerDestinationFolderPath`	String	`C:\inetpub\wwwroot\TMS`	If you would like to use a directory containing spaces, see above on using the `CMDLINE` parameter.
`PrivilegeManagerLogFile`	String	None	Optional
`PrivilegeManagerSiteName`	String	Default website	`CreateWebsite=1` must also be set to customize this.

Required Database Parameters

Delinea does not support using SQL Express in a production environment due to size and performance limitations.

Table: Required Database Parameters

Parameter	Value	Default	Notes
`DatabaseConnectionTimeout`	Integer	15	Database connection timeout in seconds.

Parameter	Value	Default	Notes
`DatabaseEnableMultiSubnetFailover`	Boolean	0	If your application connects to an AlwaysOn availability group (AG) on different subnets, setting `MultiSubnetFailover=true` provides faster detection of and connection to the (currently) active server. For more information about SqlClient support for AlwaysOn availability groups, see SqlClient support for High Availability, Disaster Recovery.
`DatabaseEnableSslEncryption`	Boolean	0	Whether or not to use SSL/TLS for the database connection.
`DatabaseFailoverPartner`	String	None	The name or address of the partner server to connect to if the primary server is down. Only used if `DatabaseEnableMultiSubnetFailover=1`.
`DatabaseIsUsingWindowsAuthentication`	Boolean		Whether or not to use integrated Windows authentication for MSSQL access. If enabled, before running the install, you must configure your IIS application pool to run a Windows account permission to access the `DatabaseServer`, and the `DatabaseUserName` and `DatabasePassword` will not be used.
`DatabaseName`	String	None	Database name. Is created if it does not exist. Defaults to `SecretServer` if installing SQL Express.
`DatabasePassword`	String	None	Database SQL login password. Ignored if using Windows authentication.
`DatabaseServer`	String	None	Database server hostname or IP.

Parameter	Value	Default	Notes
`DatabaseTrustServerCertificate`	Boolean	0	Only used if `DatabaseEnableSslEncryption=1` is set. Do not enable in production if you are using SSL encryption; certificate trust validation is critical to security. `Certutil.exe` can be used to diagnose untrusted certificates. When `TrustServerCertificate` is set to `true`, the transport layer uses SSL to encrypt the channel and bypass walking the certificate chain to validate trust. If `TrustServerCertificate` is set to `true` and encryption is turned on, the encryption level specified on the server is used even if `Encrypt` is set to false. The connection fails otherwise.
`DatabaseUserName`	String	None	Database SQL login username. Ignored if using Windows authentication.
`InstallSqlExpress`	Boolean	0	Whether or not to install the free SQL Express to use as a database server. If enabled, none of the other database parameters are used. We only recommended this for testing. , do not use in production due to performance limits.

Email Parameters

You can set email parameter either in the UI after installation or by using these parameters at install time.

Table: Optional Email Parameters

Parameter	Value	Default	Notes
EmailDomain	String (optional)	None	Domain for SMTP credentials. Used if `EmailUseCredentials=1`.
EmailFromAddress	String (required)	None	The "from" address to use when sending emails.
EmailPassword	String (optional)	None	Password for SMTP credentials, used if `EmailUseCredentials=1`.

Parameter	Value	Default	Notes
EmailPort	Integer	25	The TCP port used to connect to the SMTP server.
EmailServerName	String (required)	None	Hostname or IP of a SMTP server.
EmailUseCredentials	Boolean	0	Whether or not SMTP credentials should be sent when connecting.
EmailUseCustomPort	Boolean	0	Whether or not to use a custom port connecting to the email server.
EmailUserName	String (optional)	None	Username for SMTP credentials. Used if `EmailUseCredentials=1`.
EmailUseSSL	Boolean	0	Whether or not to use SSL/TLS when connecting to the email server

Single-Line Example

This example installs Secret Server and not PM, leaving variable for the database parameters:

```
ThycoticSetup.exe -q -s InstallSecretServer=1 InstallPrivilegeManager=0
DatabaseServer=<hostname> DatabaseName=<SecretServer> DatabaseUserName=<username>
DatabasePassword=<password> /l C:\temp\ss-install.log /nodetect
```

Moving Secret Server to Another Machine

This topic only applies to **Secret Server On-Premises**.

Always backup your data before performing this operation.

If you are moving/migrating Secret Server to a new machine and have installed IIS and .NET Framework as described in the Installation Guide on the new machine, you do not need to run the installer; you just need to follow the steps below:

1. If you use the "Force HTTPS/SSL" option, disable it by clicking **Configuration** from the **Administration** menu, and then click the **Security** tab, and **Edit**. You can re-enable the "Force HTTPS/SSL" option after you set up and install an SSL certificate on the new machine.

   If you are also moving the SQL Server database, be sure to create a new backup of the database, as this setting is written to it. To move the database, follow the steps in "Moving the Microsoft SQL Server Database to Another Machine" on page 101.

2. If you have configured encryption of your key using DPAPI, you will also need to turn this off before continuing with Step 3. To do so, click **Configuration** from the **Administration** menu, then click the **Security** tab. Click

Decrypt Key to *not use* DPAPI and enter your Secret Server account password.

3. Copy the folder that holds your Secret Server instance to the new computer.
4. Shut down the old web site and recycle its application pool as it is running background threads that are accessing the database.
5. Set up the new folder in Internet Information Server (IIS) as a virtual directory/application under the Default Web Site or as a separate Website. For detailed instructions, refer to "Advanced (Manual) Installation" on page 65 in the Installation Guide.
6. If your database server and credentials have not changed, skip this step. If they have changed, follow the steps below:
   a. Delete the database.config file from the secretserver folder (on the ASP.NET/IIS machine).
   b. Restart your new Secret Server website, so it is running.
   c. Browse to your Secret Server URL \ dbconnectionreset.aspx `http:\\secretserverurl\dbconnectionreset.aspx` and you will be prompted to enter your new database connection details.
   d. Enter your new SQL Server and the account information.
   e. Click **Next** and the site will connect to the new database. Your site is now pointing the new database.
7. Activate the licenses for the new server by going to the **Licenses** page.
8. If you are using certs, remember to set them on your new IIS, and then browse to Secret Server over HTTPS and re-enable force HTTPS if this was set on the original machine.
9. Re-enable DPAPI if this was disabled in the earlier step.


If you are moving Secret Server web application from Windows Server 2008 to 2012 AND your Secret Server is below version 8.5, make sure that:

- .Net extensions 3.5 and ASP.Net 3.5 when adding the IIS role on the new server.
- Change the Secret Server application pool to 2.0 and recycle the application pool after running the installer.

Licensing


This topic only applies to **Secret Server On-Premises**.

The Secret Server licensing model allows for scalability and enhanced core functionality in the form of edition enhancements (Professional, Premium Edition) and user packs. Licenses can be purchased for these items as follows:

- **Users:**Secret Server ships with one free license for a single user. Additional user licenses can be purchased through https://delinea.com.
- **Support:** Support licenses allow you to receive technical assistance from the Secret Server support team, and software updates for installed versions of Secret Server. To be eligible for updates, the number of support licenses and user licenses must match.

After installation is complete, you need to enter your licenses using the Getting Started Wizard or from the Licenses Administration page. Entering your licenses allows you to add more users and enables additional features in Secret Server.

For more information, see "Adding, Activating, Converting, and Deleting Licenses" below.

Licensing Limited Mode

If you fail to activate your licenses, your system is placed in limited mode, which prevents the following actions:

- AD sync
- Creating and editing secrets
- Importing secrets
- Creating and editing secrets
- Web services (mobile applications)

Adding, Activating, Converting, and Deleting Licenses

This topic only applies to **Secret Server On-Premises**.

This section explains how to add and activate Secret Server licenses (both online and offline) how to delete licenses, and how to convert from a trial license.

For more information on understanding Secret Server licensing, see the "Licensing" on the previous page page.

Offline Activation

For offline activation, go to the License Activation Center.

Adding and Activating Secret Server Licenses Online or Offline

1. Log on to Secret Server as an administrator.
2. Go to **Admin > Licenses** in the **Setup and System Upkeep** section. The Licenses page appears.
3. Click the **Install New License** button. The Install New License popup appears:
4. Click to select **Single Entry** in the **Entry Type** selection button.

If you have numerous licenses, you can click the license **Bulk Entry** selection instead. It allows you to paste an entire licensing email or a formatted list of licenses, adding all licenses in a few clicks. For a small number of licenses, especially if you are new to the process, we recommend using single entry, which provides better feedback on what you are doing.

5. Type (or paste) the **License Name** and **License Key** for the license that you received from your account manager.
6. Click the **Install** button. The License Installed Successfully popup appears.
7. If you have another license, click the **Install Another License** button to repeat the process.
8. Click the **Continue with Activation** button. The License Activation page appears.
9. Ensure your name, email address, and phone number are present and correct.
10. If you have an internet connection and want to activate **online**:
    a. Click the **Activation Type** dropdown list and select **Online**.
    b. Click the **Activate** button. An Activation Successful popup briefly appears and then disappears, and you are returned to the Licenses page where your new license now appears. The procedure is complete. **Do not do the remaining steps**.
    c. The endpoint called by Secret Server for posting data is: https://my.thycotic.com/LicenseActivation.ashx
11. If you do not have an internet connection on the Secret Server machine and want to activate **offline**:
    a. Click the **Activation Type** dropdown list and select **Offline**. The Offline Activation section appears.
    b. Click the **Copy to Clipboard** link to copy the text in the **Request** text box.
    c. Copy the contents of your clipboard to a thumb drive or the like. Leave the Secret Server page open.
    d. On another machine that does have access to the Internet, go to the License Activation Center.
    e. Paste the copied text into the text box.
    f. Click the **Activate** button. Activation Successful! appears at the top of the page and the text box now contains the activation confirmation.
    g. Copy the entire text box contents.
    h. Return to Secret Server and paste the response into the **Response** text area.
    i. Click the **Activate** button. An Activation Successful popup briefly appears and then disappears, and you are returned to the Licenses page where your new license now appears.

For more information on activating Secret Server licenses, see the "License Activation FAQ" on the next page. Secret Server may be activated on an air gap network for both trials and licensed products. Please let your Account Manager know you will be using Secret Server on an air gap network for more information.

If you receive an error message, please take note of the error code and call the phone number contained in the message.

If an error message persists after successful activation, remove expired and invalid licenses from Secret Server by following the steps below, under **Deleting Secret Server Licenses**.

If you need help and your Secret Server has a current support license for each user license, please contact our technical support team.

For more information on Secret Server licensing and license activation, see "Licensing" on page 80 and the "License Activation FAQ" on the next page.

Converting Evaluation Licenses

If you had evaluation licenses initially and you recently purchased Secret Server, you need to remove all evaluation licenses before you install your purchased licenses. Follow the steps below, under **Deleting Secret Server Licenses**.

Deleting Secret Server Licenses

1. Log on to Secret Server as an administrator.
2. Go to **Admin > Licenses**.
3. In the **Licenses** dialog, click the **License Name** of the license you want to remove.
4. Click **Delete**. The license information will remain available to you from your account.
5. Click **OK**.
6. Verify that the selected license key has been removed from the list.

License Activation FAQ

This topic only applies to **Secret Server On-Premises**.

What happens if we find that we had more named users than licenses after activation? Will the account lock us out? The user licenses are per named individual. You can simply disable any excess users so you are within your license count–these users can be re-enabled later and all audit log information is kept.

Why is license activation required?

Activation of license keys is standard practice in the software industry. We try to focus exclusively on implementing customer requests but occasionally we must spend time on licensing especially as Secret Server goes into new geographical markets.

Is there a grace period before we must activate?

Existing customers have 30 days to activate their licenses after upgrading. New licenses must be activated immediately on adding them to Secret Server. Evaluation licenses do not require activation.

How is license activation implemented in Secret Server? Activation is per license and Web server (the combination of the two). Therefore, even if a Web server was already activated, if you bring up a new Web server, it also needs activation. The activation process gathers the name, email, and phone number of the individual activating, for internal purposes only. No other personal information is sent to Delinea.

What will happen if we don't activate our licenses?

Secret Server will go into limited mode if you do not activate your licenses. Limited mode allows you to view passwords, but many other features are disabled such as creating secrets, editing secrets, changing permissions, and using Web services. Activate your licenses to get out of limited mode.

We have several license keys. Do we need to activate each license key individually?

No, the license activation process will activate all license keys that are currently added to your Secret Server. However, additional license key for distributed engine may need to be activated individually if you receive the key after the other licenses.

What if we have been using our license keys on more than one instance of Secret Server?Secret Server software licenses (user, professional, enterprise, or enterprise plus edition licenses) may only be used on a **single production instance** of Secret Server. You may use your same licenses for a single testing (non-production) environment. If you have used your licenses on multiple production instances of Secret Server, please contact us.

What information is collected and sent during license activation?

License Activation is required for each web server that will be running Secret Server. The request and the response to or from `delinea.com` are encrypted for added security.

The following information is sent to `delinea.com` when you activate:

- Name (user entered)
- Phone number (user entered)
- Email (user entered)
- All licenses (license name, license key)
- Hardware hash of each Web server

The following information is one-way hashed or omitted before it is sent to ensure it does not reveal any identifiable hardware information.

- Secret Server version
- An encrypted value to identify the instance
- Secret data or the `encryption.config` file (both omitted).
- The data that is gathered for is for contacting you if there is a licensing issue. Delinea will not sell or distribute the information provided during activation. The only information available to Delinea staff is the contact information for technical support and customer service.

Our Secret Server does not have outbound access to the Delinea.com Web site. Can activation be done while offline?

Yes, there is an offline option for activating licenses. See "Adding, Activating, Converting, and Deleting Licenses" on page 81.

If we have trouble activating our licenses, what should we do?

1. If your Secret Server is currently supported, with an equal number of current support licenses and user licenses, our technical support team can help you. Please contact us.
2. If an error message persists after successful activation, remove expired or invalid licenses from Secret Server by clicking the license name and then deleting it (the license information will remain available to you from your account.

My Server is a VM that moves to different hardware often. Will this cause me to need to reactivate over and over?

You do not need to reactivate over and over. When you activate, you can use Secret Server for a year without needing to reactivate, regardless VM hardware changes. However, if your machine name changes as well as your hardware, you will need to reactivate. If you are using a version older than 7.8.000000, you must reactivate when the VM moves.

Every time a customer switches Secret Server Databases the Secret Server license activation requirement appears. Is this now expected behavior?

Secret Server activation is now based on the Secret Server database name instead of the host name, hence, you can add or change nodes to a Secret Server cluster without reactivating. Activation requires both the license and Secret Server database. Therefore, even if you already activated a Secret Server Web server, when you install another, it too requires activation.

The activation process gathers the name, email, and phone number of the individual activating Secret Server for internal use. Delinea gathers no other personal information.

Viewing Your License

To view your current license for Secret Server cloud version navigate to **Admin > Settings > Cloud Subscriptions**. All your active subscriptions are listed here. Select the related subscription from the list to view the details, such as the number of users, secrets, sites, engines, session monitoring storage and session monitoring secrets with their used and remaining capcity.

The purchased version is not yet available in Secret Server On-Premises. Contact the purchaser at your company to inquire which version of Secret Server you purchased (this is listed in the executed order form from Delinea).

Prerequisites

"Secret Server Major Browser Support " below

"System Requirements for Secret Server" on the next page

Secret Server Major Browser Support

Secret Server can accommodate most major browsers available today. This article covers each major browser and version supported by Secret Server, as well as support for the copy-to-clipboard feature.

For the best security, always keep your browser updated to the latest version.

Browser	Supported Secret Server Versions	Copy-to-Clipboard Support
Chrome	Version 10.8 and later	Version 10.8 and later
Edge	Version 10.8 and later	Version 10.8 and later
Firefox	Version 10.8 and later	Version 10.8 and later
Internet Explorer	Not supported	Not supported
Safari	Version 10.8 and later	Version 10.8 and later
Opera	Version 10.8 and later	Version 10.8 and later

Language Support

Secret Server natively supports the following languages:

- English
- French
- German
- Japanese
- Korean
- Mandarin, simplified
- Mandarin, traditional
- Portuguese

Using Chrome to Access Secret Server

The Chrome extension prompts users for "tabs" permissions, which Chrome uses to detect your browsing history, including what tabs you open and the URLs they open to. Secret Server uses the tabs permission function only to clear passwords on exit. No history is recorded.

System Requirements for Secret Server

This topic applies to **Secret Server On-Premises** and standalone **Secret Server Cloud**.

Most of this topic applies to Secret Server On-Premises, but the distributed engine requirements also apply to Secret Server Cloud.

All cores are physical unless otherwise noted.

Important: Please read the notes at the bottom of this article.

Minimum Requirements for Basic Deployments

Web Server	Database Server
4 CPU Cores	4 CPU Cores
16 GB RAM	16 GB RAM
25 GB Disk Space	100+ GB Disk Space
Windows Server 2016-2022	Windows Server 2016-2022
IIS 7 or newer (64-bit applications only)	SQL Server 2016-2022
.NET 4.8 or newer	Collation SQL_Latin1_General_CP1_CI_AS on both SQL Server and the Secret Server database.

Minimum Requirements for Advanced Deployments

For organizations deploying significant discovery, session recording, event pipelines, syslog, or many distributed engines:

Web Server	Database Server
8 CPU Cores	8 CPU Cores
16 GB RAM	16 GB RAM
25 GB Disk Space	100+ GB Disk Space
Windows Server 2016-2022	Windows Server 2016-2022
IIS 8 or newer (64-bit applications only)	SQL Server 2016-2022
.NET 4.8 or newer	Collation SQL_Latin1_General_CP1_CI_AS on both SQL Server and the Secret Server database.

Note: Windows Server 2022 is supported by Secret Server 11.0 or later. It may work with earlier versions, but that has not been officially confirmed.

SQL Server Is Usually Required

Secret Server requires an instance of SQL Server for the database backend and is installed by the setup.exe file, if missing. The SQL Server database will require a SQL account with *db_owner* permission to complete the installation.

Delinea does not support using SQL Express in a production environment due to size and performance limitations.

Recommended Session Recording Requirements

Session Recording Requirements

See "Basic Session Recording Requirements " on page 1230 and "Advanced Session Recording Requirements" on page 1229.

Distributed Engine and RabbitMQ MinimumRequirements

Distributed Engines	RabbitMQ Messaging Server
4 CPU Cores	4 CPU Cores
4 GB RAM	4 GB RAM
25 GB Disk Space	40 GB Disk Space

> Although these may be the minimum requirements, the way you utilize the product will influence the resource needs, and you may require additional resources.

For scalability and reliability, Delinea strongly recommends using RabbitMQ. MemoryMQ is an easier but less capable alternative and can be used for trials and proof of concepts but should not be used for production environments.

Further adjustments to system requirements for both RabbitMQ and distributed engines are at the discretion of Delinea Professional Services engineers.

The latest Distributed Engine supported version is always the latest current version. Older versions can also be run if they are up to date or if the version used is just behind the latest. If there are breaking changes, an update will be forced.

For RabbitMQ, there is not an officially supported set of versions. Installation is only supported using the latest RabbitMQ Helper, which will install the latest versions. Any version of RabbitMQ available since the introduction of durable queues will work. If you wish to install an older version you can at your own discretion. Please note they might be prone to certain errors in functionality.

System Requirements for Virtual Machines and Processors

A vCPU is a virtualized CPU core assigned to a virtual machine (VM) from the physical CPUs available on the host server. The following are some key points. Please see What is a vCPU and How Do You Calculate vCPU to CPU? for details.

Key points:

- vCPUs allow multiple VMs to share the physical CPU cores on a host through time-slicing and context switching.
- The ratio of vCPUs to physical cores determines CPU oversubscription. An oversubscribed CPU is shared among too many vCPUs, leading to potential performance issues.
- There is no fixed ratio for calculating optimal vCPU to physical core mapping. It depends on the specific workloads and resource demands of each VM.
- Monitoring CPU utilization for the VMs and host is crucial to identify performance bottlenecks and adjust vCPU allocations accordingly.
- Using too many vCPUs unnecessarily can increase licensing costs for some software that charges per vCPU.
- It is important to understand application resource needs, monitoring performance metrics, and adjusting vCPU allocations to strike the right balance between oversubscription and performance for cost optimization.

Notes

> Important: This section contains caveats potentially having a significant effect on any installation.

- For the highest scalability and reliability, Delinea recommends using RabbitMQ. MemoryMQ is an easier but less capable alternative and can be used for trials and proof of concepts but **should not** be used for production environments. Two exceptions are very small deployments and customers that do not use open-source

software for compliance reasons.

- The use of Server Core for Secret Server installations is not recommended; All QA and testing is based non-core versions of Windows Server.
- To comply with Microsoft licensing requirements, there is an additional constraint on which Microsoft Windows Server version you can use as the RDS server for session connector.

  If you use Microsoft User Client Access Licenses (CALs), you cannot use Windows Server 2019. You must use Windows Server 2012 or 2016. If you use Microsoft Device CALs, you can use any supported version of Windows Server.
- Secret Server requires that Microsoft SQL Server and its database be set to the collation SQL_Latin1_General_CP1_CI_AS. See Microsoft SQL collation requirements and check your server collation settings before upgrading.
- System Requirements apply to both physical and virtual machines.
- For best performance, we recommend using dedicated (clean) servers for hosting Delinea products.
- If .NET or IIS features are not already installed on the web server, the Delinea Installer will add and configure them automatically.
- If SQL is not already installed on a database server, the Delinea installer can setup SQL Express on the web server; however, SQL Express is intended for trials and sandbox environments **only**. Delinea **does not support it** because of performance issues due to the memory and product limitations.
- A link to Microsoft documentation on the use and limitations of SQL Express can be found at: https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2017.
- Installing Secret Server with Azure SQL: Currently, we do not recommend using Secret Server with Azure SQL when the Web host and the Azure SQL instance are in different datacenters. According to Microsoft, applications, such as Secret Server, that use frequent, high-volume, ad hoc queries use substantial response time on network communication between the application and Azure SQL database tiers. Thus, network latency with many data access operations across datacenters can become an issue.
- Unsupported Web Servers: Small Business Server (SBS), The Essentials Edition, Any client OS, domain controllers, SharePoint servers.
- Secret Server Cloud requires an on-premise machine to use a distributed engine.
- SQL launchers do not support SSMS 18.0 or higher.
- Discovery scanning for Windows Server 2016 scheduled tasks requires that either the Secret Server node or the distributed engine that is executing the scan must run on Windows Server 2016 or later. This is due to changes in Windows Server 2016 API used for scheduled task dependency scans.

Compatibility issues with Microsoft libraries are likely if the distributed engine or Secret Server node operates on a version of Windows Server that is older than the version on the target server it is attempting to scan, for instance, a DE on 2016 scanning a machine running 2019. To ensure successful scans, we recommend that the DE or node runs on the same or a newer version of Windows Server as the target machine.

- AWS RDS: Currently, we do not recommend using Secret Server with AWS Relational Database Service when the Web host and the SQL instance are in different datacenters. Applications, such as Secret Server, that use frequent, high-volume, ad hoc queries depend on fast network communication response time between the application and SQL database. Thus, network latency with many data access operations across datacenters can become an issue.
- Secret Server requires the application pool to have the "load user profile" setting enabled. Secret Server will report a critical alert to notify admins if this setting is not enabled.
- Supported Web browsers: See "Secret Server Major Browser Support " on page 85.

Proxied Environments

This topic only applies to **Secret Server On-Premises**.

If your Secret Server has outbound access through a proxy, its web.config must be modified to specify the proxy configuration.

If Secret Server is also clustered and has multiple worker roles enabled, the web.config must be updated for each Secret Server in the cluster.

Microsoft has more information on this.

The other option in a clustered environment is to specify a remote site for the data upload, and upload data through a Distributed Engine. If the distributed engine's host server is also behind a proxy, however, the engine's Delinea.DistributedEngine.Service.exe.config must be modified similarly to the web.config in order to specify the proxy settings.

For Secret Server v10.4 or later, the web-proxy.config can be uncommented and updated to specify the proxy settings.

For Secret Server v10.3.000015 or earlier, you must add proxy-related XML to the web.config file immediately following the file's closing \</configSections\> tag, as depicted here:

```
</configsections>
    <system.net>
        <defaultproxy enabled="true" usedefaultcredentials="true">
             <proxy
usesystemdefault="false" proxyaddress="https://proxy:port" bypassonlocal="true"/>
        </defaultproxy>
    </system.net>
<configuration type="thycotic.foundation.configuration, thycotic.foundation">
```

Using Webnode with Proxied Environments

If using a webnode you will need to add the following code:

```
<system.net>
```

```
    <defaultProxy configSource="web-proxy.config" />
</system.net>
```

DE Configuration

> Please note it is suggested the customer create an exception in the proxy for both webnodes and DEs to bypass since the configuration files will be overwritten with product updates and changes will be need to be implemented again. There is a FR for proxy settings to retain, but as of 9/21/2022 it had not been implemented.

When Secret Server and distributed engines are behind a proxy certain settings need to be added to webnodes and DEs is they exist in the environment.

To use with the Distributed Engine through a proxy, you will need to add proxy info to Thycotic.DistributedEngine.Service.exe.config between </system.serviceModel> and located in the C:\Program Files\Thycotic Software Ltd\Distributed Engine\ folder on the distributed engine. You may need to refer to the below article for other proxy related settings.

Element (Network Settings)

```
<system.net>
    <defaultProxy>
        <proxy usesystemdefault="true" />
    </defaultProxy>
</system.net>
```

You will need to restart the DE service afterwards and the setting will need to be reapplied after any Distributed Engine upgrade.

Webnode Configuration

Main Proxy settings are stored in the web-proxy.config file in the Secret Server folder on each webnode. Microsoft's article on Proxy configuration explains all settings.

Element (Network Settings)

A few examples below:

Example #1

```
<?xml version="1.0" encoding="utf-8" ?>
<defaultProxy enabled="true">
<proxy
    usesystemdefault="true"
    proxyaddress="http://192.168.1.1:8080"
    bypassonlocal="true"
  />
</defaultProxy>
```

Example #2

```
<defaultProxy enabled="true">
    <proxy proxyaddress–"http://proxy.domain.com:80" bypassonlocal-"true" / >
</defaultProxy>
```

Now the following files need to be edited to point to the web-proxy.config file.

- web-embeddedRole-backgroundScheduler.config
- web-embeddedRole-backgroundWorker.config
- web-embeddedRole-engineWorker.config
- web-embeddedRole-messageBroker.config
- web-embeddedRole-sessionRecordingWorker.config

The code used in these files can be as follows:

```
<system.net>
    <defaultProxy configSource="web-proxy.config" />
</system.net>
```

Placement of this setting may affect connection. I have confirmed success when the code is placed before the section.

These will need to be edited after each update until the aforementioned FR is implemented keeping proxy settings.

RabbitMQ and Secret Server

> This topic only applies to **Secret Server On-Premises**.

Secret Server uses RabbitMQ as its message bus or broker to facilitate efficient and reliable message traffic between various components. This asynchronous message-based system ensures that operational instructions and data are passed back and forth seamlessly, enabling tasks such as password changes, discovery, and other background operations to be processed efficiently. RabbitMQ's robust framework supports clustering and high availability, making it ideal for large-scale deployments. All messages transmitted via RabbitMQ are encrypted, ensuring secure communication. Using RabbitMQ, Secret Server can handle large volumes of messages, maintain high performance, and ensure that critical operations are executed reliably.

Clearing RabbitMQ Message Queues

> This topic only applies to **Secret Server On-Premises**.

Some users note that older RabbitMQ message queues in Ready state are not clearing as expected, so messages accumulate. To clear the message queues, use the procedure below.

1. On the machine where RabbitMQ is installed, download the utility for removing old RabbitMQ queues.
   `SHA1(RMQ_QueueRemoval.zip)= B8EE3CD2488AF2D7A42421B870EB8041434245C8`

```
SHA256(RMQ_QueueRemoval.zip)=
B9AF3BF51B0E1E6E937830A6CF0974D3546183B78E1E86F6C8563E5E7243146A
```

2. Extract the zip file.
3. Open Windows PowerShell.
4. Navigate to the directory where you extracted the zip file.
5. Load the file by typing the following command:

   ```
   . .\RMQ_QueueRemoval.ps1
   ```

6. Run the commands below in the order shown:
   a. `ShowAllQueues`
   b. `ShowQueuesNoConsumer`
   c. `DeleteQueuesNoConsumer`

Installing RabbitMQ

> This topic only applies to **Secret Server On-Premises**.

Overview

What is RabbitMQ?

RabbitMQ is a robust message queuing software package that Secret Server uses to communicate with its distributed engines. For detailed information about RabbitMQ, go to the RabbitMQ website.

Why Install It?

RabbitMQ is an enterprise-ready alternative to MemoryMQ. While MemoryMQ is sufficient for basic and prototyping installations, RabbitMQ is the preferred messaging framework when the need for greater reliability and clustering arises.

> For the highest scalability and reliability, Delinea recommends using RabbitMQ. MemoryMQ is an easier but less capable alternative and can be used for trials and proof of concepts but **should not** be used for production environments. Two exceptions are very small deployments and customers that do not use open-source software for compliance reasons.

RabbitMQ and Encryption

All data sent from or read by Secret Server from RabbitMQ is encrypted. If you would like to add SSL despite the data already being encrypted, please follow the Advanced installation of RabbitMQ with TLS use case. Please note that Delinea Support can help with non-SSL installations. For SSL installation, configuration, troubleshooting, and RabbitMQ clustering, please contact Delinea Professional Services to learn more about our Professional Services rates.

Downloading Delinea's RabbitMQ Helper

Please go to Delinea RabbitMQ Helper to download the most recent version.

Prerequisites

> Secret Server only supports RabbitMQ on Windows operating systems. Secret Server is incompatible with the RabbitMQ Federation plugin.

RabbitMQ requires:

General Requirements

- Windows Server 2008 or higher with PowerShell v3 support.
- Nodes hosting RabbitMQ need a minimum of 4 GB RAM.
- Nodes hosting RabbitMQ should have at least 128 MB of memory available at all times.
- Disk space is not an issue, but it should not go below 50 MB (default value), especially if you host RabbitMQ on the same server as Secret Server.
- Minimum of 2 vCPUs. This is an **absolute minimum** otherwise installation fails without much useful feedback to troubleshoot. We strongly recommend 4 vCPUs or more.
- Ports 5672 (non-SSL) or 5671 (SSL) opened on the machine and firewall.

SSL Certificate

- A server certificate of the PFX type and a root authority certificate of the CER type.
- The PFX certificate should have:
  - A name that matches the RabbitMQ Fully qualified machine name.
  - If you plan on making a RabbitMQ cluster, add DNS names (SANs) to your certificate.
  - Your certificate must be a RSA certificate. CNG is not supported and will cause the installation to fail.
- If you do not have an internal PKI and prefer not to use a public certificate, you can use a self-signed certificate.

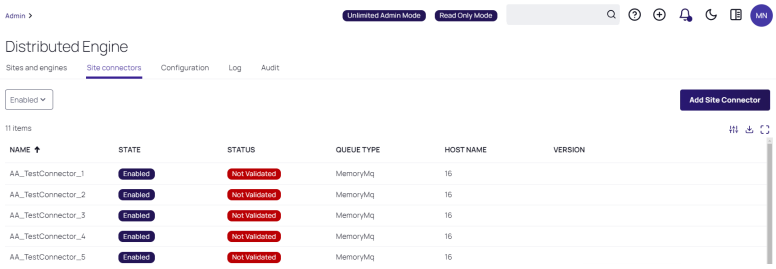
> Delinea will not assist with creating or troubleshooting self-signed certificates.

Installation

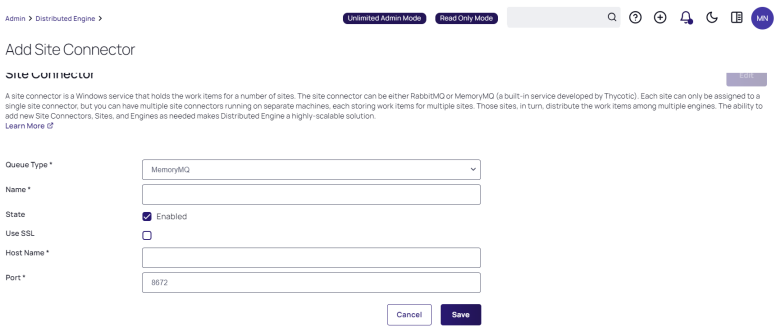
Task 1: Secret Server

In the Secret Server UI:

1. Navigate to **Admin** > **Distributed Engine**.
2. Access the **Site Connectors** tab and select **Add Site Connector**:



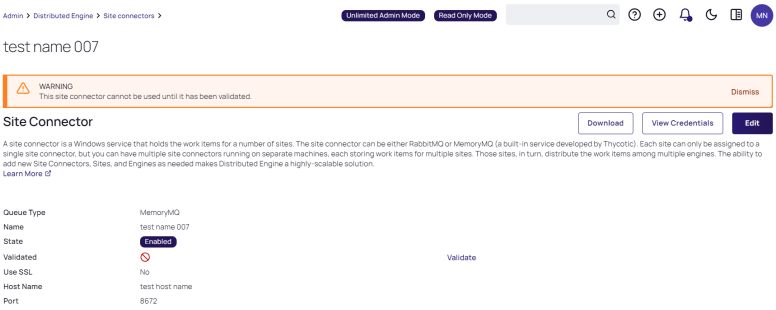
3. On the **Add Site Connector** page, select either **RabbitMQ** or **MemoryMQ** in the **Queue Type** drop-down list. If at least 3 RabbitMQ nodes have been set up in a clustering setup, choose MemoryMQ (see the Cluster section in the RabbitMQ Helper documentation for more information):



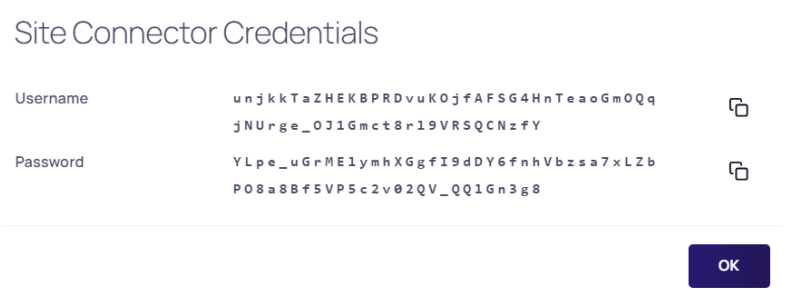
4. Type a name for your new site connector in the **Name** text box.
5. Select the **Enabled** check box.
6. Type the host name of the machine where you plan to install RabbitMQ, in the **Host Name** text box.

> The Engines need to be able to resolve this host name or the connection will fail. Inbound firewall rules must be created on the machine that is hosting the connector as well.

7. Type either port 5672 (non-SSL) or 5671 (SSL) in the **Port** text box.
8. Click the **Save** button.
9. After the site connector is created, click the site connector's link. The **Site Connector Details** will page appear:



10. Select the **View Credentials** button to retrieve the automatically generated credentials. The **Site Connector Credentials** pop-up will appear:



    You can ignore the informational message that the connectivity has not been validated for now, as you will be doing so after you install RabbitMQ on the host you have selected.

11. Select the copy icons for both the **User Name** and **Password** values, to copy and store them for use in the next section.
12. Select the **OK** button.

Task 2: RabbitMQ Host

See RMQ Helper Installation.

Troubleshooting

Please refer to RabbitMQ Helper.

Secret Server Cloud Offboarding

This topic only applies to standalone **Secret Server Cloud**.

Privacy Policy

See the Delinea Privacy Policy for details on how your data is safeguarded.

Data Protection

At all times, even after your license expires, Delinea maintains strict security controls of your data. Only a tiny group of operations staff can access it and only with approval from at least one other person. You can get a SOC2 audit report for your instance, which requires an NDA, by emailing the RFP Helpdesk. The NDA is required because proprietary Delinea information is present in the report.

Your Data When Your Subscription Ends

Once your subscription ends:

1. Your instance enters a two-week grace period.
2. After the grace period expires, the instance is scheduled for deletion two weeks in the future. Your instance is inaccessible but all data is retained.
3. Once the deletion date arrives, your instance is eligible for deletion and can be deleted at any time at Delinea's discretion. In most cases, Delinea retains customer data substantially beyond the deletion date.

If desired, you can request that Delinea deletes your instance right away, and we will promptly respond. **Once deleted, your instance is not recoverable and your data is gone.**

SQL Server and Secret Server

> This topic only applies to **Secret Server On-Premises**.

Secret Server utilizes Microsoft SQL Server as its backend database to store and manage all its data securely and efficiently. SQL Server provides a robust and scalable platform for handling the extensive data operations required by Secret Server, including storing secrets, audit logs, user information, and configuration settings. The database is configured to use the SQL_Latin1_General_CP1_CI_AS collation to ensure compatibility and optimal performance. Secret Server supports various editions of SQL Server, including Express, Standard, and Enterprise, allowing organizations to choose the edition that best fits their needs. For high availability and disaster recovery, Secret Server can leverage SQL Server's advanced features such as mirroring, clustering, and AlwaysOn Availability Groups. Additionally, Secret Server can be configured to use Integrated Windows Authentication (IWA) for secure and seamless access to the SQL Server database.

Choosing a SQL Server Edition to Use with Secret Server

> This topic only applies to **Secret Server On-Premises**.

Choose the Microsoft SQL Sever edition to work with Secret Server that best supports the functionality you wish to achieve.

The brief guide below should help you decide which licensing model best suits the needs of your organization.

SQL Server Express Edition

SQL Server Express is a free version of SQL that is sufficient to run most of the functionality within the Secret Server application itself.

However, advanced functionality like mirroring and clustering is not available.

> Delinea does not support using SQL Express in a production environment due to size and performance limitations.

SQL Server Standard Edition

SQL Server Standard provides most of the functionality administrators typically want, including the most common type of mirroring, and clustering up to two cluster nodes.

SQL Server Enterprise Edition

SQL Server Enterprise provides all of the functionality found in the Standard Edition, plus the ability to cluster up to eight nodes and to perform asynchronous mirroring.

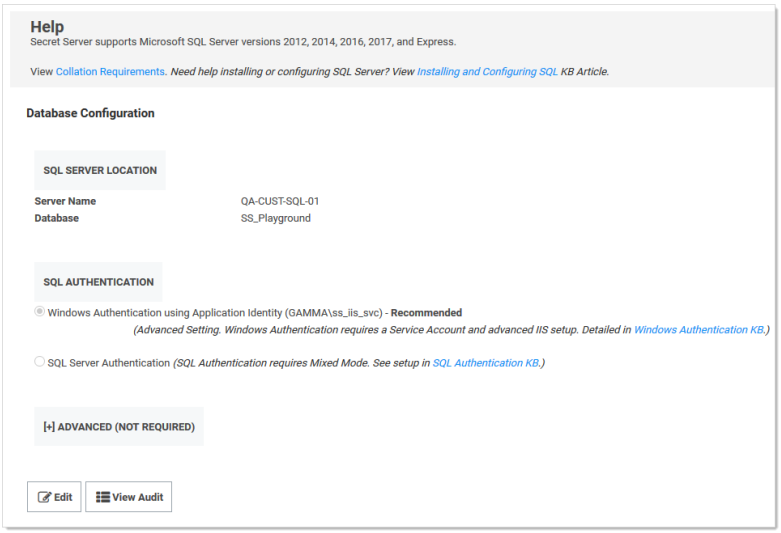
For more information on the different editions of SQL Server, see the Microsoft SQL Server 2016 Licensing Guide.

Enabling SQL Server Encryption

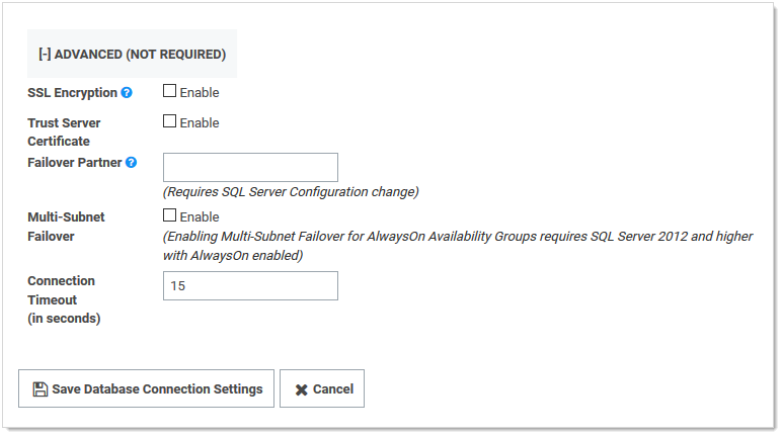
> This topic only applies to **Secret Server On-Premises**.

Administrators can enable end-to-end encryption with the SQL database by using an Encrypted connection. This is a feature that is built into Microsoft SQL Server and Secret Server supports. To enable encryption:

1. Go to **Admin > See All**. The admin panel appears.
2. Type Database in the **Search** text box and select **Database**. The Database Configuration page appears:



3. Click the **Edit** button.
4. Click the **Advanced (Not Required)** link. A new section appears:



5. Click to select the **SSL Encryption** check box.
6. Click the **Save Database Connection Settings** button.

> SQL Server must be pre-configured to support encryption. This Microsoft TechNet article explains how to configure the SQL Server environment for encryption. The SSL encryption used for communicating with SQL Server is either 40 or 128 bit, depending on the Windows operating system used.

> Using this setting can adversely affect performance. See this TechNet article for additional information.

Installing and Configuring SQL Server

> This topic only applies to **Secret Server On-Premises**.

For step-by-step instructions on how to install SQL 2016, see "SQL Server 2016 Standard Edition Installation" on page 111.

Secret Server requires Microsoft SQL Server as the back-end database. All editions, including the Express version, of 2012-2019 are supported.

> Delinea does not support using SQL Express in a production environment due to size and performance limitations.

Setting up SQL Server requires:

- Installing SQL Server
- Creating a SQL Account
- Configuring database access in Secret Server
- Installing SQL Server

If you are using SQL Express make sure to get the edition with tools that will include SQL Management Studio. Follow the link in the "SQL Server 2014 Express Edition Installation" on page 103 .

Before installing the SQL server, please make sure that the "no count" SQL Server connection property is turned off. If this setting is turned on, no Secret Server jobs will execute.

Creating a SQL Account

SQL Authentication

The fastest method to get started with Secret Server is to create a SQL Authentication account. Follow the instructions in the Database section of the "Basic (Automatic) Installation" on page 68.

For troubleshooting and configuring SQL installation on a different server that the application server see "SQL Server Authentication Configuration" on page 102 article.

Windows Authentication

A more advanced way to have Secret Server access the SQL server would be through a service account and using Windows Authentication. Because of the requirement of a service account and added IIS settings, we only recommend this for non-evaluation setups. See instructions in "Configuring Integrated Windows Authentication" on page 383.

Configuring Database Access in Secret Server

Once the account has been created and SQL server installed with the MSI. The third step of the Web installer will ask for database access information.

SQL Location

- **Server Name or IP**: If it is a local machine the server name will be (local) or localhost for the default instance, or if a named instance such as SQL Express it would be `localhost\SQLExpress`. If you are unsure, copy the value from the "Server name" text box when connecting through SQL Management Studio.
- **Database Name**: If you have created a database, enter the name. If you have given the SQL account dbCreator permission, enter a database name for Secret Server to create.

SQL Authentication

- **SQL Server Authentication**: Implies a SQL account has been created that exists only with SQL Server. The account will need to be dbOwner on the database or need dbOwner permission to create the database. This is recommended for quickest setup. For more detailed information and troubleshooting see "Installing and Configuring SQL Server" on the previous page .
- **Windows Authentication**: The identity of the application pool will access the database. This requires a domain Service account that has been granted access to run ASP.Net and the database. This is an advanced setting that is not recommended for evaluations. Follow the instructions on using a service account in "Configuring Integrated Windows Authentication" on page 383.

We suggest to setup the database maintenance to regularly backup the transaction log for the Secret Server database to prevent system down issues. Once the log fills up, it may cause Secret Server to go down. Check Managing Full SQL Server Transaction Logs for more details.

Moving the Microsoft SQL Server Database to Another Machine

This topic only applies to **Secret Server On-Premises**.

This article only applies if your MS SQL Server database is only for Secret Server.

Follow the steps below for moving MS SQL Server database for Secret Server.

Task 1: Backing up and Restoring the Database

To back up your Secret Server installation:

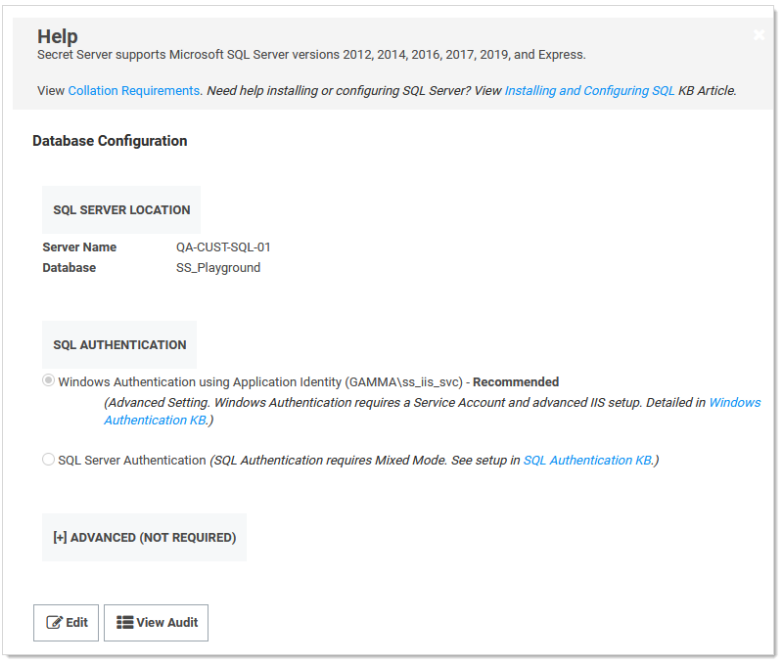
1. Enable the maintenance mode.
2. Stop the Secret Server site in Internet Information Server (IIS) to prevent any changes to the database.
3. Navigate to the directory where Secret Server is installed.
4. Copy the folder (holding the application) to your back up location.
5. Open your SQL Server Management Studio.
6. Right click the database your Secret Server is running on, and select **Tasks > Backup**.
7. Click the **Add** button. You are prompted to enter a file path for the `.bak` file. This can be the final destination (not recommended) or a temporary one (for later moving to a back up location).
8. Make sure SQL Server has permissions for this location. That is, create (if needed) and or grant access to the account that will access the database (see the "Installation" on page 65 for account creation instructions). See "Running the IIS Application Pool As a Service Account" on page 60 (Task 2) for details.
9. Copy the resulting database backup file (`.bak`) to your backup location.

You can also automate steps 2-4 using the command: `osql -S myserver\SQLEXPRESS -E - Q "BACKUP DATABASE SECRETSERVER TO DISK = 'c:\backup\ss.bak'.`

We recommend taking the old database offline after all steps are complete.

Task 2: Connecting Secret Server to the New Database

1. Restart your Secret Server website in IIS.
2. Log on Secret Server as a local admin.
3. Navigate to `https://<your_SS_URL>/Setup/Database`. The Database Configuration page appears:



> The setting here are stored in `C:\inetpub\wwwroot\Secretserver\database.config`. You can back that file up to revert or simply return to this page to reset the connection again. See the Privilege Manager documentation if you need to change its configuration too.

4. Click the **Edit** button. The page becomes editable.
5. Type your new SQL Server location (server name) and database.
6. Click the **Save Database Connection Settings** button, and the site will connect to the new database.
7. Your site is now pointing the new database.

> To roll back changes and restore the original database, complete both tasks again to move the database back to the original database server.

> If you are also moving the Secret Server application to another server, see "Moving Secret Server to Another Machine" on page 79 for more information.

> The steps in Task 2 should be performed for each web node in a deployment. Making the change on one node only will not propagate to the other nodes.

SQL Server Authentication Configuration

> This topic only applies to **Secret Server On-Premises**.

SQL Authentication requires:

- Creating a new SQL account
- Enabling mixed mode
- Enabling named pipes and SQL Browser a non-local SQL Server

> For instructions on Creating the SQL account or Installing SQL Server see "Installing and Configuring SQL Server" on page 99 article.

Enabling Mixed Mode

1. Connect to SQL Server in SQL Management Studio.
2. Right click on the instance node and select **Properties**.
3. Go to the **Security** tab.
4. In the **Server Authentication** section, select **SQL Server and Windows Authentication Mode**.
5. Click the **Ok** button.
6. Restart the SQL Server, by right clicking on the instance node and selecting **Restart**.

> If your SQL server is running on a separate machine, you need to turn on named pipes and SQL browser to ensure the SQL server can be accessed from an external machine.

Enabling Named Pipes and SQL Browser

1. Open SQL Server Configuration Manager.
2. Click the **SQL Server Network Configuration** node.
3. Select **Protocols for MSSQLSERVER**.
4. Enable the following:
   - Shared memory
   - Named pipes
   - TCP/IP
5. Enable **SQL Browser**.
6. Click to select the **SQL Server Services** node.
7. Right click **SQL Server Browser** and select **Start**.

SQL Server 2014 Express Edition Installation

> This topic only applies to **Secret Server On-Premises**.

Overview

> Delineadoes not support using SQL Express in a production environment due to size and performance limitations.

SQL Express is a free edition of SQL and is available for use with Delinea products. The following steps walk you through setup and configuration for SQL Server 2014 Express Edition as an example. For the most up to date resources on installing SQL see Microsoft SQL Technical Documentation for more information.

At the completion of this article you will have:

- Installed a basic stand-alone instance of SQL Server 2014 Express with the minimum features necessary for SQL Server. This includes SQL Server Management Studio and other tools.
- Created a database in SQL for your Delinea product
- Created a new SQL Server user login for your SQL database

> This document uses Delinea's Secret Server product as example in the instructions, but the same steps apply for Privilege Manager advanced installs.

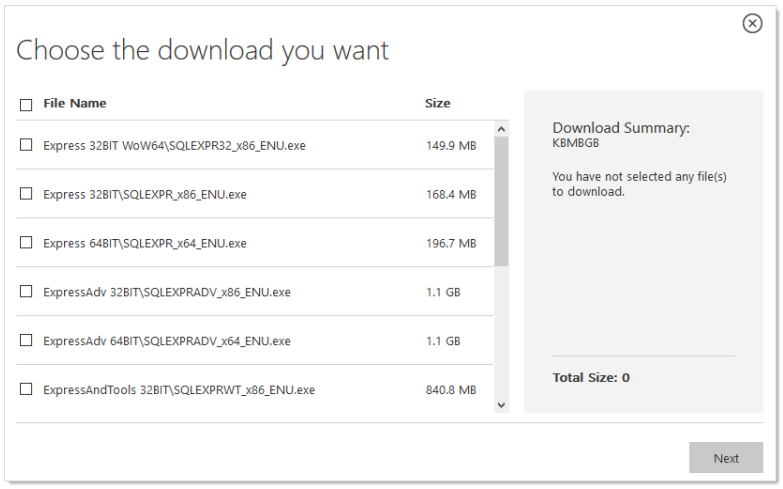
Procedures

Downloading SQL Server Express with Tools

If you plan to use SQL Server Express, we strongly recommend downloading the package that includes **Tools**. This also installs SQL Server Management Studio that allows you to connect to the database directly and gives access to server settings.

Procedure:

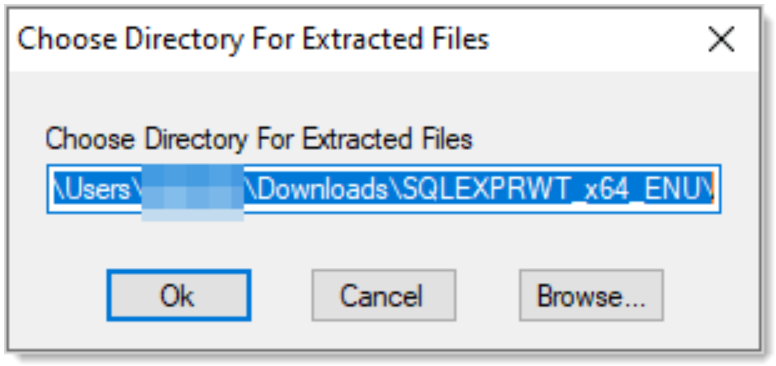
1. Go to the SQL Server 2014 Express download page.
2. Click the **Select Language** list box and select **English**.
3. Click the **Download** button. A popup page appears:



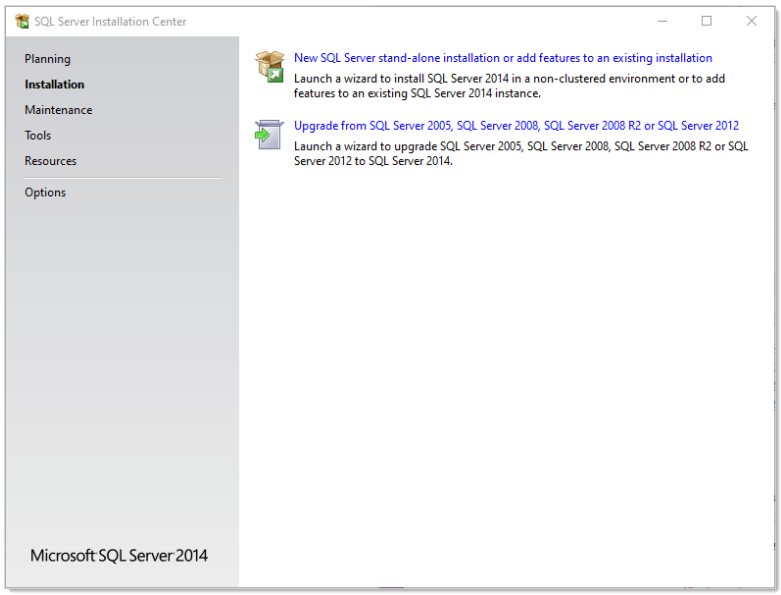
4. Click to select the following check boxes (you may need to scroll down):
   - **ExpressAndTools 64BIT\SQLEXPRWT_x64_ENU.exe**
   - **MgmtStudio 64BIT\SQLManagementStudio_x64_ENU.exe**
5. Click the **Next** button. `SQLEXPRWT_x64_ENU.exe` and `SQLManagementStudio_x64_ENU.exe*` download to your computer.

Installing SQL Server Express 2014

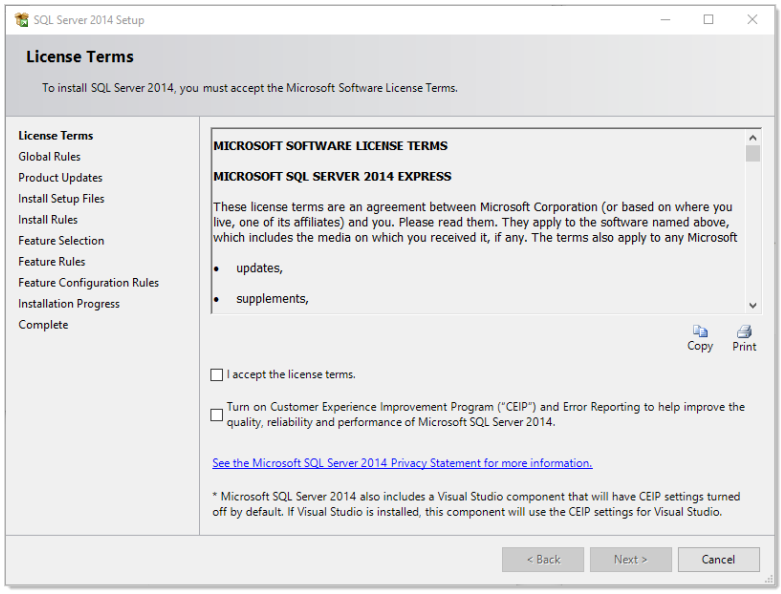
1. If necessary, download and install the latest version of .NET Framework. See Microsoft .NET Framework 4.8 offline Installer for Windows for the latest version as of when this topic was written. If you have already installed Secret Server, you have already done this.
2. Double click the `SQLEXPRWT_x64_ENU.exe` you downloaded to run it. The User Account Control appears.
3. Click the **Yes** button. The Choose Directory... dialog box appears:



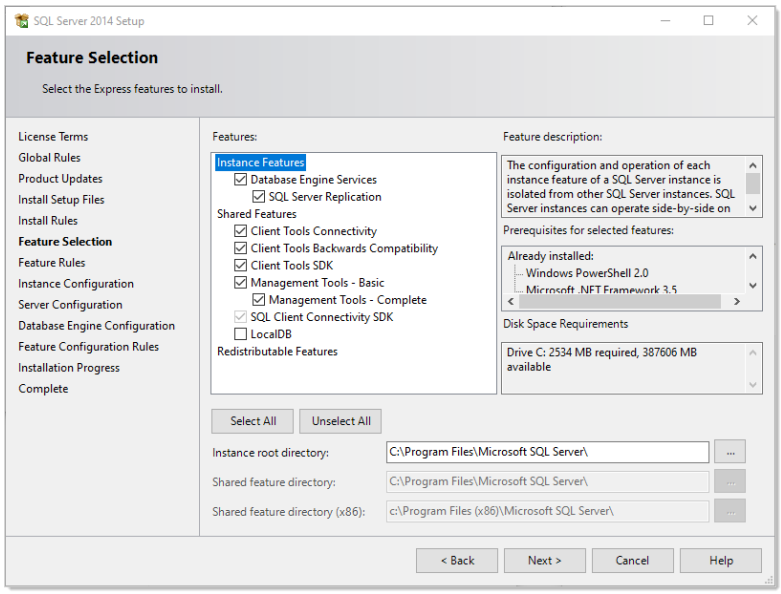
4. Click the **OK** button. The files are extracted to that location, and the SQL Server Installation Center appears:



5. Click the **New SQL Server stand-alone...** link. The License Terms wizard page appears:



6. Click to select the **I accept the license terms** check box.
7. Click the **Next >** button. The installation processes four pages with no input from you and stops on the Feature Selection page:

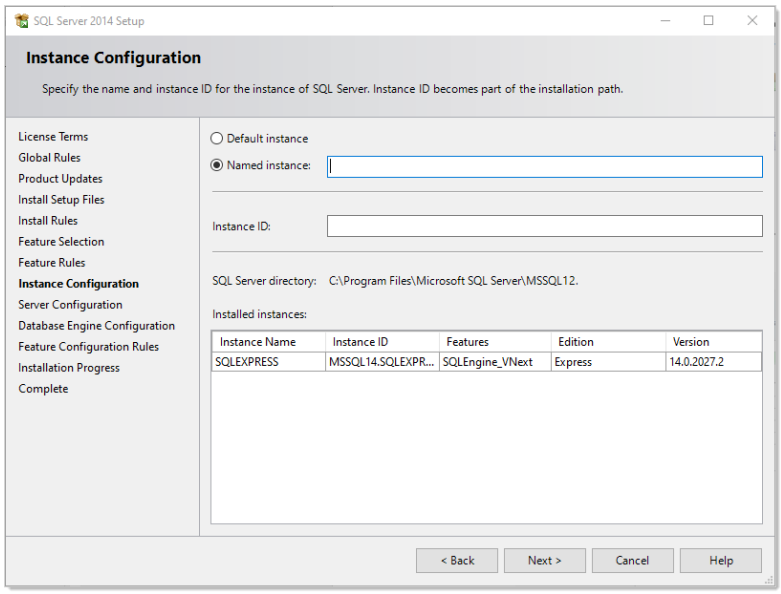


8. Ensure that the **Database Engine Services** and **Management Tools – Basic** check boxes are selected. Leave the others as is.

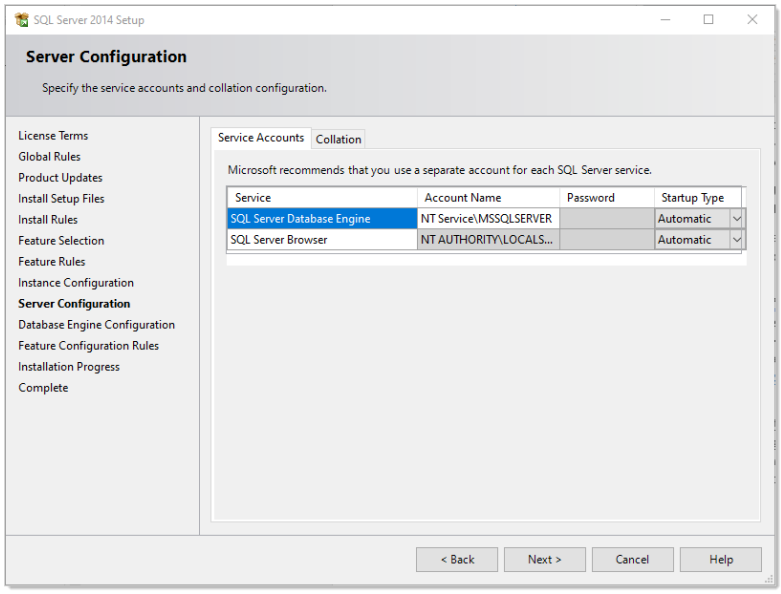
   A SQL Server instance is isolated from other SQL Server instances. SQL Server instances can operate side-by-side on the same computer.

   Management tools include Management Studio support for the database engine and SQL Server Express, SQL Server CLI (SQLCMD), SQL Server PowerShell provider, and the distributed replay administration tool.

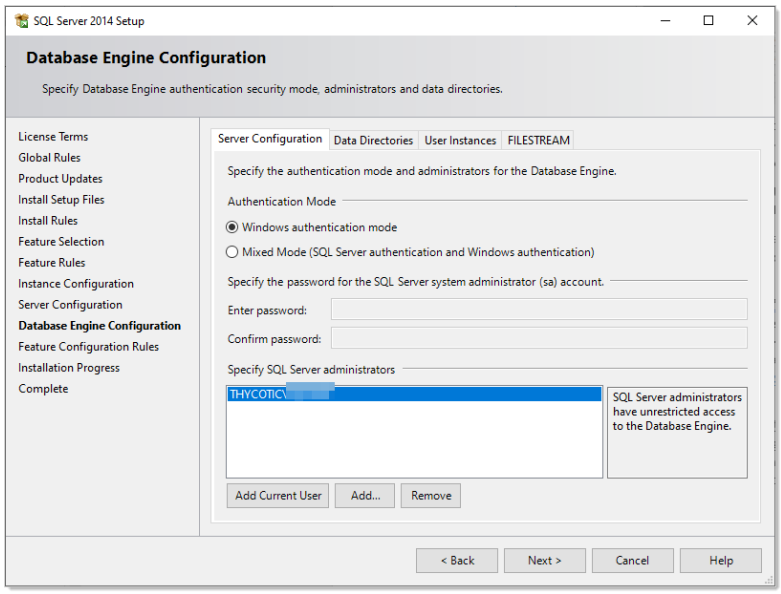
9. Click the **Next >** button. The installation processes one page with no input from you and stops on the Instance Configuration page:



10. **Ether** click to select the **Default instance** selection button, which uses an already present instance called SQLEXPRESS.
11. **Or** type your desired name in the **Named instance** text box.
12. Type your instance ID in the **Instance ID** text box. We chose `MySQLInstance`. The instance ID will become part of the installation path.
13. Click the **Next >** button. The Server Configuration page appears:



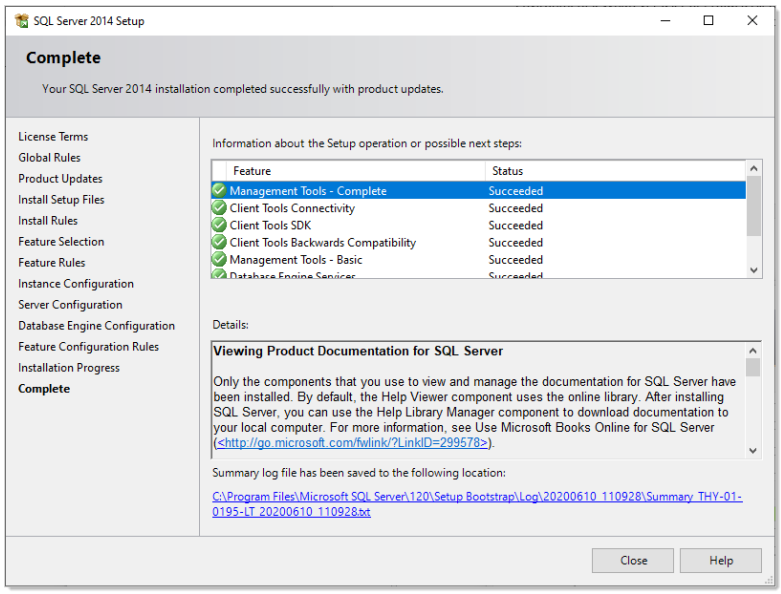
14. Leave the page as is, and click the **Next >** button. The Database Engine Configuration page appears:



15. You have the choice to select either **Windows Authentication Mode** or **Mixed Mode**. Click to select the option that works best for your environment:
    - **Mixed Mode (for easiest configuration)**: This mode is required if you intend on using a SQL Server account to authenticate Secret Server to your SQL Server instance. **We recommend using mixed mode if you are setting up a test or demo environment**. Selecting this option will also require you to set a password for the SQL Server system administrator (sa) account. See **Adding a SQL Server User** below for instructions on adding more users.
    - **Windows Mode (recommended for best security)**: This mode prevents SQL Server account authentication. We recommend using Windows mode for production environments. Whatever user or group assigned will have administrative access to your SQL instance. According to best security practices, limit this number to as few users as possible. Only choose this if you have experience and require this for a specific issue–we do **not** recommend SQL Server Express for production accounts.

    > If choosing **Windows Mode** you will also need to "Running the IIS Application Pool As a Service Account" on page 60 later in the installation process.

16. If you selected mixed mode, which you almost certainly did, type your SQL Server system administrator (sa) account password in the **Enter password** and **Confirm password** text boxes. The password must meet Microsoft's definition of a strong password. Click the **Help** button and search for "Database Engine Configuration - Account Provisioning" if you what to find out what that is. A 16 character mixture of lower and uppercase letters and numerals works fine.
17. Your user account should already be shown in the **Specify SQL Server administrators** text box. If not, click the **Add Current User** button.
18. Click the **Next >** button. The Installation Progress page appears and SQL Server Express is installed. This can take awhile. Eventually, the Complete page appears:



19. Click the **Close** button.

Creating the SQL Server Database

To install Secret Server, the Delinea installer creates the SQL database for you if it does not exist and if the user account has permission to create a new database, which requires the dbcreator server role.

If not using the Delinea Installer, use the following steps to create a database manually through SQL Server Management Studio:

1. Open SQL Server Management Studio.
2. Connect to your SQL Server instance.
3. Right click the **Databases** folder and select **New Database...** The New Database page appears.
4. Type a name for your database in the **Database Name** text box.
5. Click the **OK** button.

Adding a SQL Server User

According to security best practices, limit the number of users with access to your SQL database as much as possible. Use the following instructions to add a SQL Server account for Secret Server to use to access the SQL database:

1. Open SQL Server Management Studio.
2. Connect to your SQL Server Database.
3. Expand the **Security** folder.
4. Right-click the **Logins** folder and select **New Login...**
5. Select a method of authentication:

- **SQL Server Authentication**: Use this option to create a new SQL Server account (this requires mixed mode to be enabled). To create the account, enter a new username and password and then deselect the **Enforce Password Policy** check box to prevent the account from expiring.
- **Windows Authentication**: Use this option to add access to SQL Server for an existing Windows account. To add the account, enter the login name or click **Search** to find the account. It is recommended to use a domain account rather than a local Windows account.

6. Click **User Mapping** in the left menu.
7. Click to select the check box next to your Secret Server database.
8. In the **Database Role Membership** window, click to select the **db_owner** check box.
9. Click the **OK** button.

SQL Server 2016 Standard Edition Installation

This topic only applies to **Secret Server On-Premises**.

Overview

The following steps walk you through setup and configuration for SQL Server 2016 Standard Edition as an example. For the most up to date resources on installing SQL see Microsoft SQL Technical Documentation for more information.

At the completion of this article you will have:

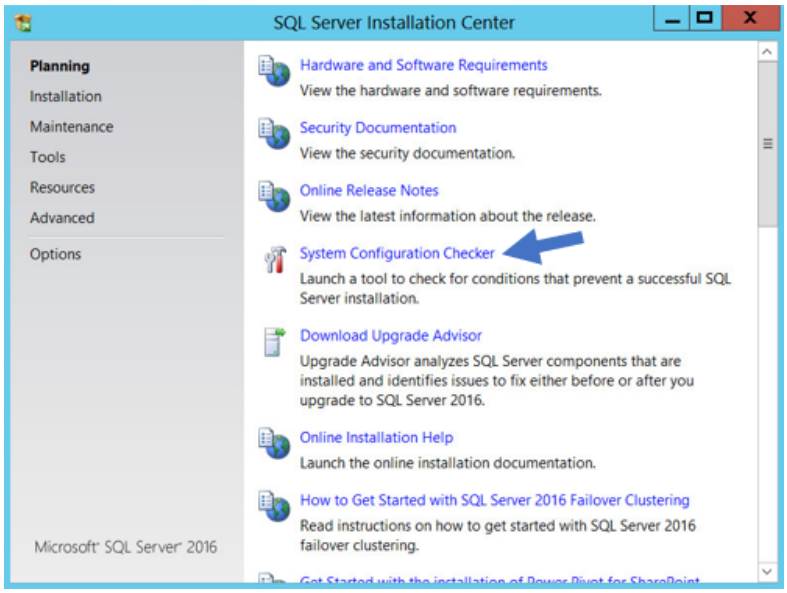
- Installed a basic stand-alone instance of SQL Server 2016 Standard with the minimum features necessary for SQL Server.
- Installed SQL Server Management Studio for managing the local database.
- Created a database in SQL for your Delinea product
- Created a new SQL Server user login for your SQL database

This document uses Delinea's Secret Server product as example in the instructions, but the same steps apply for Privilege Manager advanced installs.

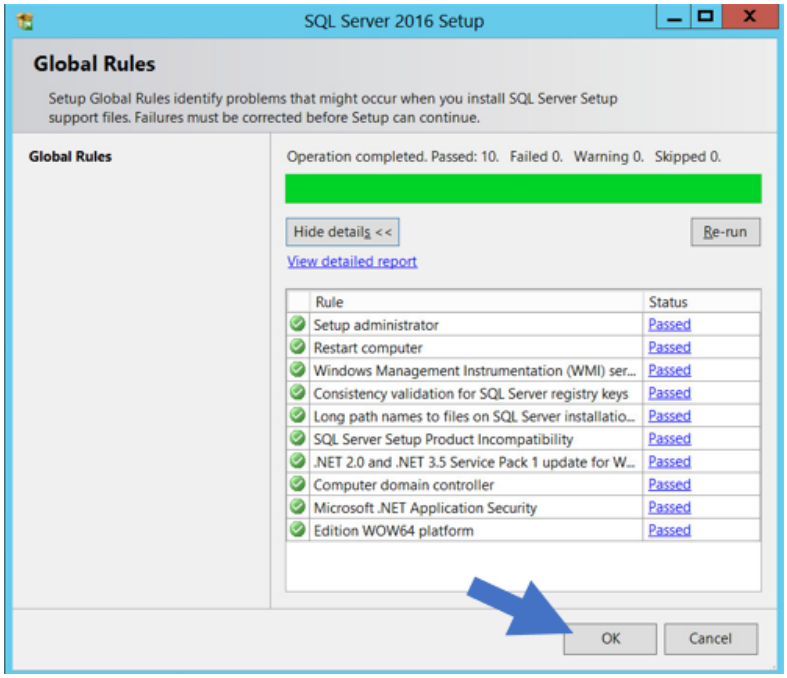
Procedures

Installing SQL Server 2016

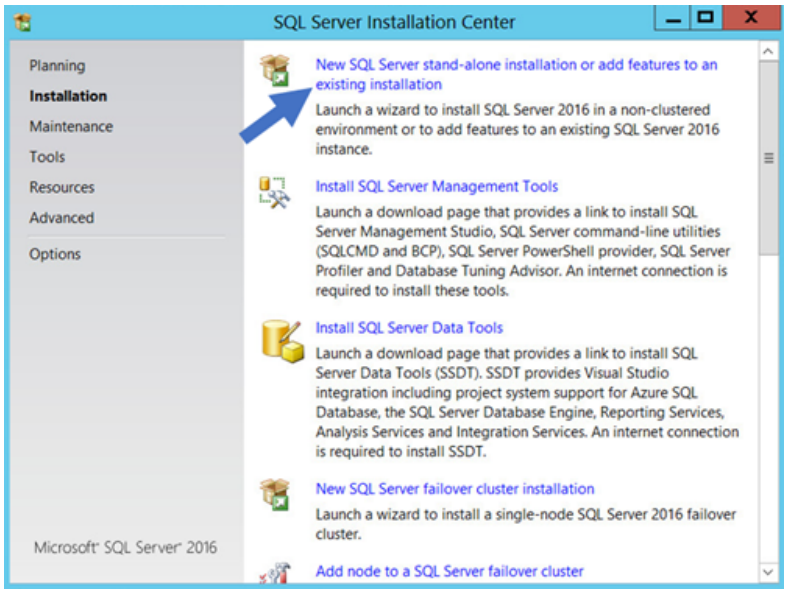
1. Launch the SQL Server installer from CD or file download. The SQL Server Installation Center opens to the Planning window:



2. Click the **System Configuration Checker** link. This runs a tool that checks for conditions on your server that could prevent SQL Server from installing.
3. When the tool launches, click the **Show details** button. A successful scan should look like the one shown below. If you encounter any issues, look at the detailed report, resolve the reported issues, and rerun the scan.

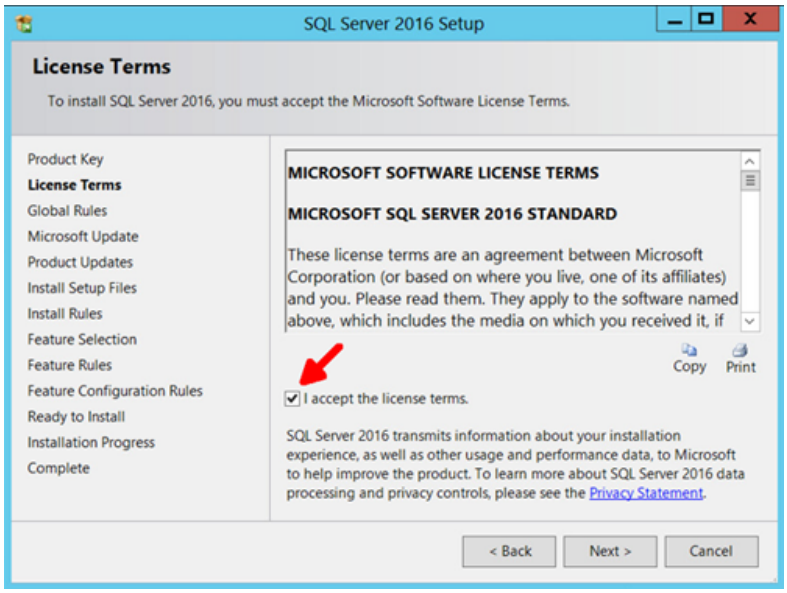


4. Click the **OK** button when done to return to the "SQL Server Installation Center" window.
5. In the SQL Server Installation Center window, click the **Installation** link. The Installation Window appears:

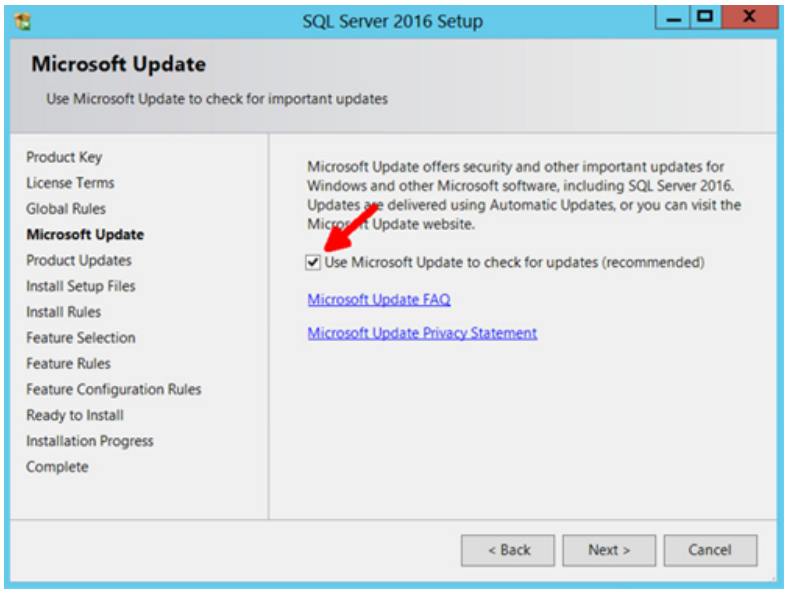


6. Click **New SQL Server stand-alone installation…** link. The Product Key page appears:

7. Click to select the **Enter the Product Key** selection button.
8. Type your product key in the **Enter the Product Key** text box.
9. Click the **Next >** button. The License Terms page appears:

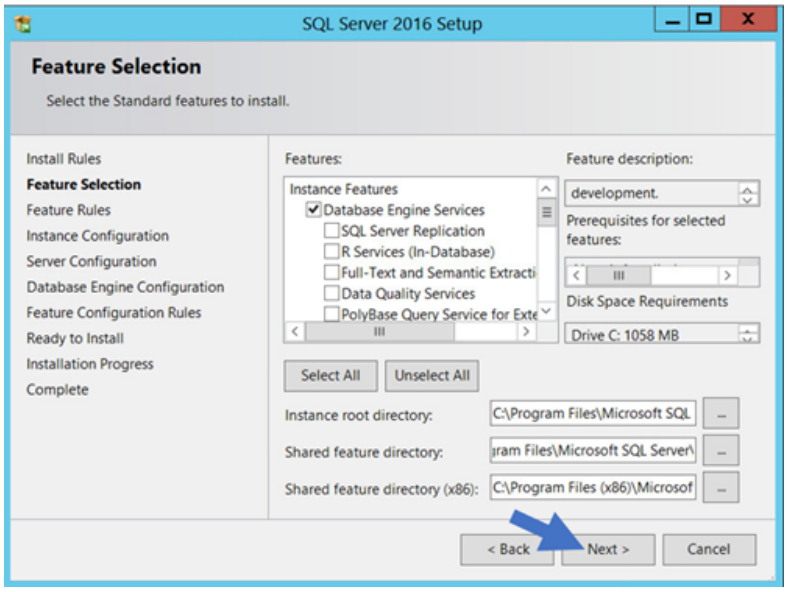


10. Click to select the **I accept the license terms**. check box.
11. Click the **Next >** button. The Global Rules page appears (not shown) after the rule check runs.
12. Click the **Next >** button. The Microsoft Update page appears:

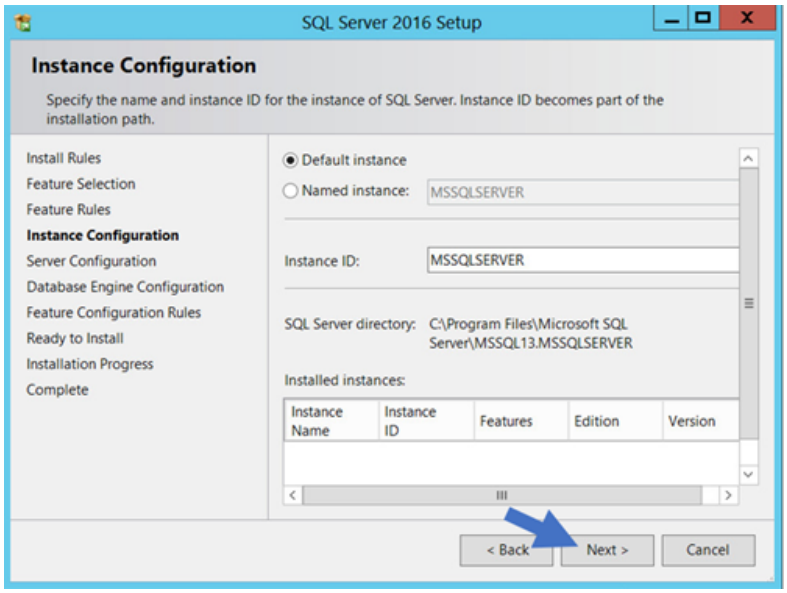


13. Click to select the **Use Microsoft Update...** check box to check for updates (recommended), unless your software update process does not use automatic updates from Microsoft
14. Click the **Next >** button twice to bypass the Product Updates page. The Install Setup Files page appears.
15. Wait for the installation to complete.
16. Ensure that all operations pass.

17. Click the **Next >** button twice to bypass the Install Rules page. The Feature Selection page appears:

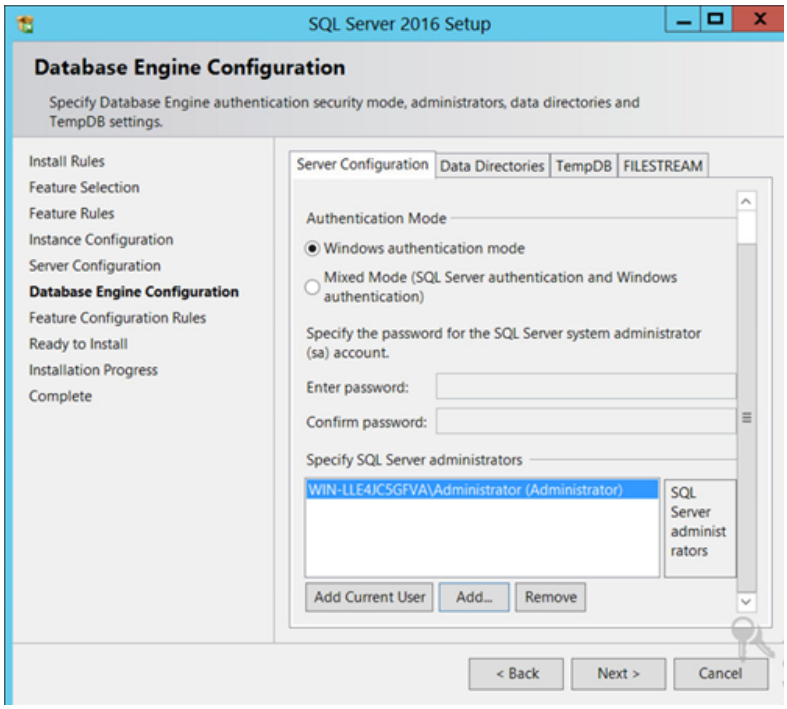


18. Ensure the **Database Engine Services** check box is selected. This is the only feature necessary for Secret Server. Unless you are using Geo-Replication, you can leave everything else unchecked. Leave the directory locations unchanged.
19. Click the **Next >** button twice to bypass the Feature Rules page. The Instance Configuration page appears:



20. Ensure the **Default Instance** selection button is selected.
21. Type a name for your SQL Instance in the **Instance ID** text box.

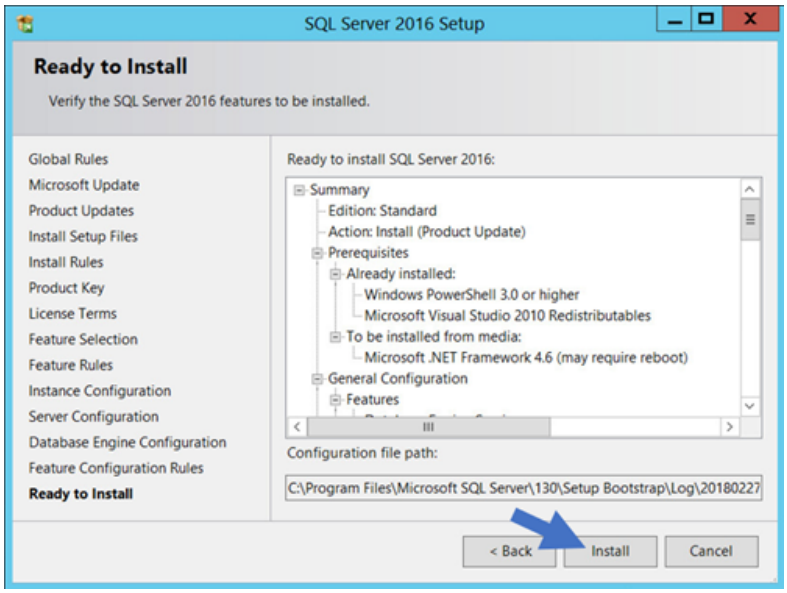
22. Click the **Next >** button twice to bypass the Server Configuration page. The Database Engine Configuration page appears:



23. You have the choice to select either **Windows Authentication Mode** or **Mixed Mode**. Select the option that will work best for your environment:
    - **Mixed Mode (for easiest configuration)**: This mode is required if you intend on using a SQL Server account to authenticate Secret Server to your SQL Server instance. We recommend using mixed mode if you are setting up a test or demo environment. Selecting this option will also require you to set a password for the SQL Server system administrator (sa) account. See **Adding a SQL Server User** below for instructions on adding more users.
    - **Windows Mode (recommended for best security)**: This mode prevents SQL Server account authentication. We recommend using Windows mode for production environments. Whatever user or group assigned will have administrative access to your SQL instance. According to best security practices, limit this number to as few users as possible.

> If choosing **Windows Mode** you will also need to "Running the IIS Application Pool As a Service Account" on page 60 later in the installation process.

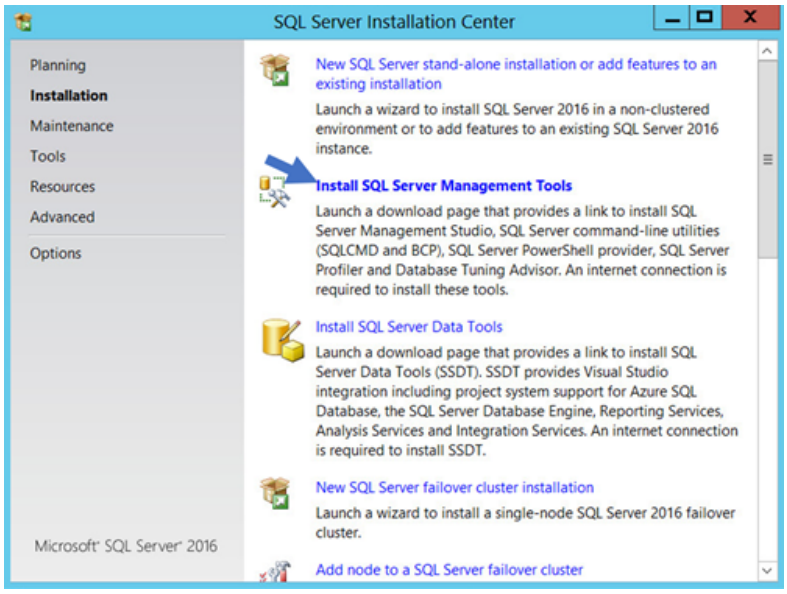
24. You can leave the options in the remaining tabs at their default values or change the file locations in the **Data Directories** and **TempDB** tabs if you wish to store the database and log data in a different drive or directory.
25. Click the **Next > button** twice to bypass the Feature Configuration Rules page. The Ready to Install page appears:



26. Click the **Install** button.
27. Wait for installation to complete. This may take several minutes.
28. Click the **Close** button.

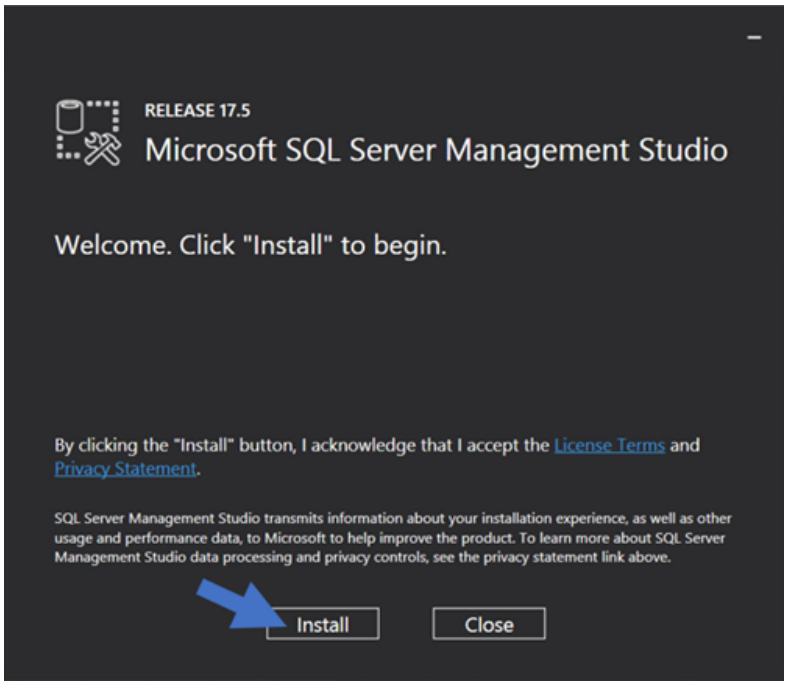
Installing SQL Server Management Studio

1. In the "SQL Server Installation Center" window, click the **Installation** menu item. The Installation page appears:

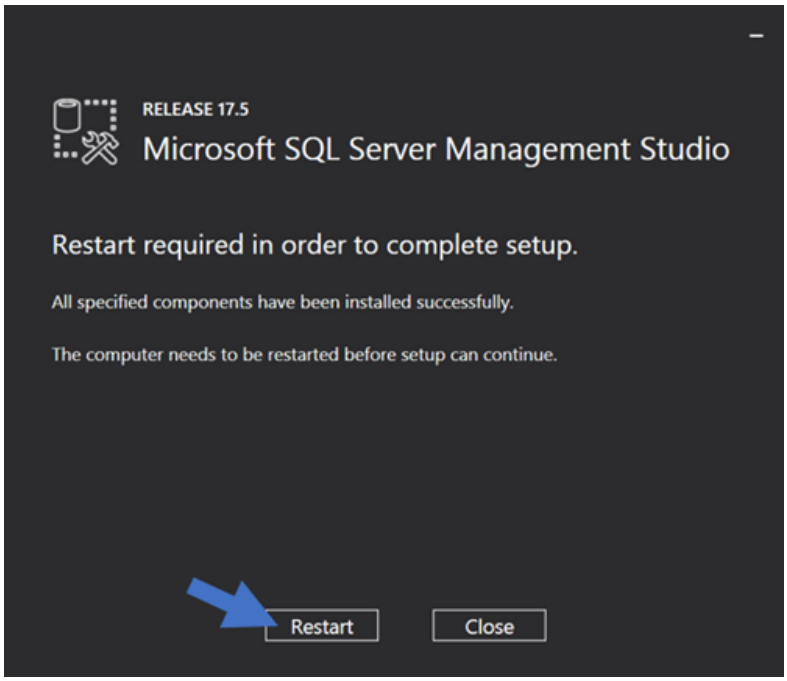


2. Click the **Install SQL Server Management Tools** link.

3. Wait for the Web page to load then click the **Download SQL Server Management Studio...** link. A file downloads.
4. Run the downloaded file (varies by browser). The SQL Server Management Studio installer starts.



5. Click the **Install** button.
6. Wait for the installer to complete. This may take several minutes.



7. Click the **Restart** button if prompted. Otherwise, click the **Close** button.
8. Close "SQL Server Installation Center."

Creating the SQL Server Database

To install Secret Server, the Delinea installer creates the SQL database for you if it does not exist and if the user account has permission to create a new database, which requires the dbcreator server role.

If not using the Delinea Installer, use the following steps to create a database manually through SQL Server Management Studio:

1. Open SQL Server Management Studio.
2. Connect to your SQL Server instance.
3. Right click the **Databases** folder and select **New Database...** The New Database page appears.
4. Type a name for your database in the **Database Name** text box.
5. Click the **OK** button.

Adding a SQL Server User to Secret Server

According to security best practices, limit the number of users with access to your SQL database as much as possible. Use the following instructions to add a SQL Server account for Secret Server to use to access the SQL database:

1. Open SQL Server Management Studio.
2. Connect to your SQL Server Database.
3. Expand the **Security** folder.

4. Right-click the **Logins** folder and select **New Login...**
5. Select a method of authentication:
   - **SQL Server Authentication**: Use this option to create a new SQL Server account (this requires mixed mode to be enabled). To create the account, enter a new username and password and then deselect the **Enforce Password Policy** check box to prevent the account from expiring.
   - **Windows Authentication**: Use this option to add access to SQL Server for an existing Windows account. To add the account, enter the login name or click **Search** to find the account. It is recommended to use a domain account rather than a local Windows account.
6. **Either**, if you have already created the database for your Delinea product:
   a. Click **User Mapping**in the**Select a page**list box.
   b. Click to select the check box for the database in the**Users mapped to this Login**list.
   c. Click to select the**db_owner**check box in the**Database role membership...**list.
7. **Or**, if you have not yet created the database:
   a. Click**Server Roles**in the**Select a page**list box.
   b. Click to select the**db_creator**check box.
8. Click to select the check box next to your Secret Server database.
9. In the **Database Role Membership** window, click to select the **db_owner** check box.
10. Click the **OK** button.

SQL Server Performance Improvement

This section provides insights and recommendations for improving SQL Server performance, focusing on reducing blocking and optimizing database operations.

Recommendations

Snapshot Isolation

We highly recommend enabling both `ALLOW_SNAPSHOT_ISOLATION` and `READ_COMMITTED_SNAPSHOT` in your database settings. This configuration allows transactions to work with a consistent snapshot of the data, significantly reducing blocking. While writers will still block other writers, most read operations will not be blocked.

To enable snapshot isolation, execute the following commands:

```
ALTER DATABASE MyDbName SET ALLOW_SNAPSHOT_ISOLATION ON;
ALTER DATABASE MyDbName SET READ_COMMITTED_SNAPSHOT ON;
```

This change allows sessions in the `READ_COMMITTED` isolation mode (the default) to automatically use snapshot isolation, requiring no application changes.

Using READ_UNCOMMITTED

For many application workflows that do not require strict transaction isolation, using `READ_UNCOMMITTED` mode can eliminate most blocking. This mode allows reading uncommitted changes, which can be suitable for scenarios where absolute accuracy is not critical.

HADR_SYNC_COMMIT Considerations

HADR_SYNC_COMMIT waits can occur during indexing operations. The ALTER INDEX ... REBUILD command is an offline operation, making the table inaccessible during the process. It is recommended to perform such operations during off-hours.

Troubleshooting

Analyzing Blocking

To get a sum of all blocking, use the following query:

```
SELECT OBJECT_NAME(o.object_id), i.name, row_lock_wait_in_ms + page_lock_wait_in_ms AS
"milliSecondsBlockedSinceRestart"
FROM sys.dm_db_index_operational_stats (DB_ID(), NULL, NULL, NULL) o
JOIN sys.indexes i ON i.object_id = o.object_id AND i.index_id = o.index_id
ORDER BY row_lock_wait_in_ms + page_lock_wait_in_ms DESC
```

This query provides a detailed view of blocking times for each index, helping identify potential bottlenecks.

Understanding SLEEP_BPOOL_FLUSH

The SLEEP_BPOOL_FLUSH wait type should generally be ignored. It is part of SQL Server's mechanism to categorize session blocking. This wait type is not necessarily indicative of a user transaction waiting.

The buffer pool is used for caching database pages. When a page is updated, it is marked as dirty and eventually written back to disk. If the disk response time exceeds 20ms, the process throttles itself. A response time of 24ms, for example, is not considered excessive.

HADR_SYNC_COMMIT Considerations

HADR_SYNC_COMMIT waits can occur during indexing operations. The ALTER INDEX ... REBUILD command is an offline operation, making the table inaccessible during the process. It is recommended to perform such operations during off-hours.

For on-hours work, consider using online index operations, which take short-lived locks, allowing the table to remain accessible. If the HADR_SYNC_COMMIT waits are still unsatisfactory, consider discussing asynchronous commit mode.

By implementing these strategies, you can enhance SQL Server performance, reduce blocking, and ensure smoother database operations.

Uninstalling Secret Server

This topic only applies to **Secret Server On-Premises**.

Following these instructions ensures there is absolutely no residue or trace of Secret Server on the server.

Uninstalling Secret Server is a quick, three-step process:

1. Delete the database.
2. Delete the virtual directory.
3. Delete Secret Server files.

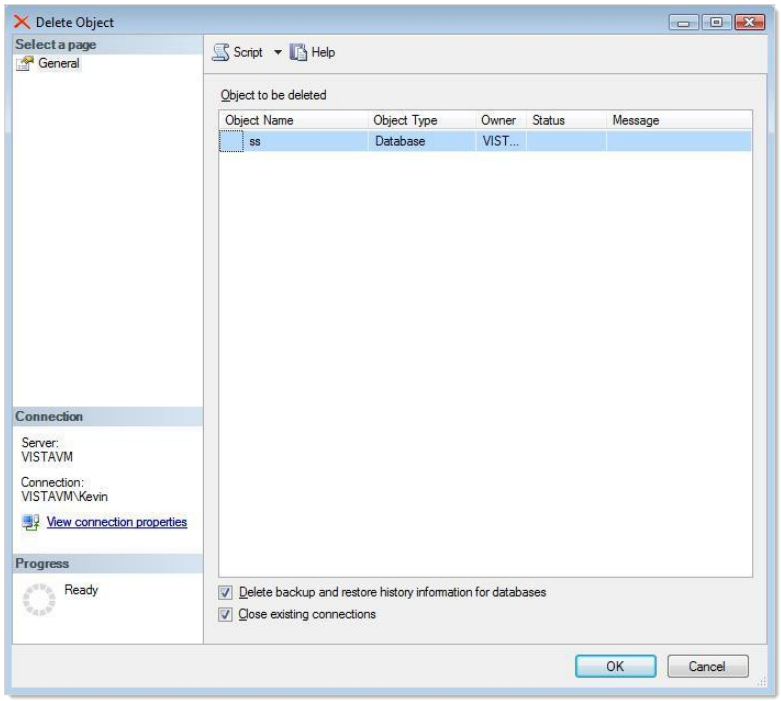
Task 1: Deleting the Database

Dropping the database deletes all of your data.

> You cannot undo this procedure once you are done. We strongly suggest backing up the installation first in case you need to restore it.

Procedure:

1. Open the Microsoft Management Studio.
2. Connect to the database.
3. Locate your Secret Server database in the object explorer, which is normally in the **Databases** folder. If necessary press F8 to show the object explorer.
4. Right click the database and select **Delete**. The Delete Object dialog appears:

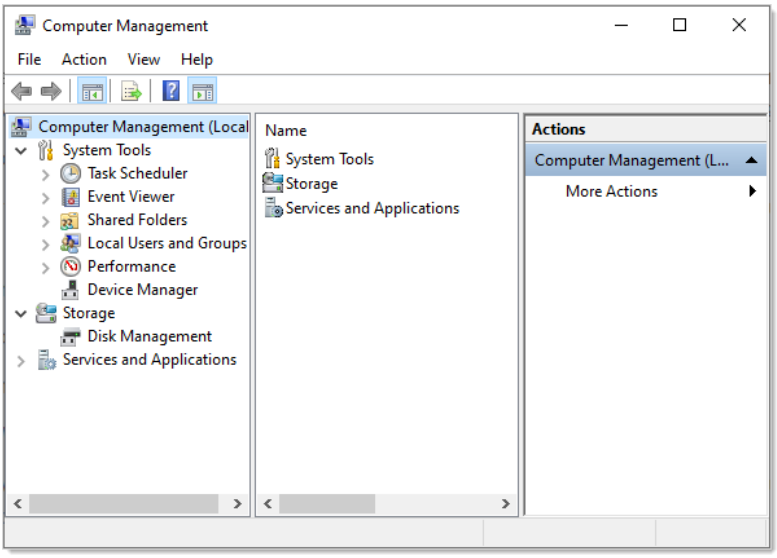


5. Ensure the **Drop Existing Connections** check box is selected. This disconnects all connections to the Secret Server database.
6. Ensure the **Delete backup and restore history information for databases** check box is selected.
7. Click the **OK** button. The database is permanently deleted.

Task 2: Deleting the Virtual Directory

If you installed Secret Server as a virtual directory, the virtual directory must be deleted first. If Secret Server is not configured as a virtual directory, skip this task.

1. In the search text box in your **Start Menu**, type `Computer Management`.
2. Click the **Computer Management** result. The Computer Management Console appears:



3. Click to expand the **Services and Applications** node.
4. Click the **Internet Information Services (IIS) Manager** node.
5. Click to expand the **Web Sites** subfolder.
6. Click to expand the Secret Server Web site.
7. Right click the virtual directory and select **Delete** or **Remove**. The directory is deleted.
8. (Optional) Delete ASP.NET's cached version of Secret Server:
   a. Open the directory `C:\Windows\Microsoft.NET\Framework\<version number>\Temporary ASP.NET Files`, substituting your ASP.NET version number.
   b. Delete the subfolder with the same name as your virtual directory.

> These files are not a security risk, but removing them eliminates any evidence that Secret Server was installed.

Task 3: Deleting Secret Server Files

> The `encryption.config` file is crucial to restoring any backup. Ensure this file is backed up if you may want to restore Secret Server.

1. Locate the directory where Secret Server is was installed.
2. Click to select it.
3. Press **<Shift>** + **<Delete>** to **permanently** delete the files. Holding shift bypasses the recycle bin. Secret Server is now permanently removed from the system.

> Even "permanently" deleted files can sometimes be recovered with special tools. If that is a concern, we suggest using a file shredding application to delete the folder.

Upgrading

> This topic only applies to **Secret Server On-Premises**.

"Ensuring Upgrade Security" below

"Manual Rolling Upgrade" on page 140

"Minimizing Upgrade Downtime" on page 128

Secret Server and Secret Server Cloud .NET Framework 4.8 Mandatory Upgrade

"Upgrading Secret Server" on the next page

"Upgrading Secret Server Without Outbound Access" on page 149

"Upgrading Secret Server with Web Clustering" on page 126

Ensuring Upgrade Security

> This topic only applies to **Secret Server On-Premises**.

We take the following measure to ensure the upgrade is secure:

Secret Server on-Premises upgrades are packaged in a wrapper zip file for delivery. These upgrade zip files are named with the release version, such as `version_11_1_000006.zip`. Each wrapper zip file contains two files, a security catalog file named `hashes.crt` and yet another zip file named `ss_update.zip`.

The `ss_update.zip` file holds the changes for the new Secret Server version (database and file changes). The `hashes.crt` file is a signed security catalog containing the hash of the `ss_update.zip` file, which is used to ensure the authenticity and integrity of the `ss_update.zip` file.

A hash is a long string of characters that represents a unique digital "fingerprint" of any file the hash function is run on. Any change to a hashed file causes future hashes of the altered file to differ from the original hash, proving the file is different.

When the upgrader processes `ss_update.zip`, the following occurs:

1. The `hashes.crt` file is validated to ensure that its digital signature is the original produced by Delinea.
2. The same hash function that created the original hash is run on the `ss_update.zip` file, creating a new hash.
3. The original hash is compared to the new hash, ensuring the two match and `ss_update.zip` has not been

tampered with.

4. The original wrapper zip file is inspected to ensure it does not contain any unexpected, likely malicious, files.

As a result of this inspection process, the upgrader is positive both `ss_update.zip` and `hashes.crt` are genuine and there are no foreign files present.

Upgrading Secret Server

> This topic only applies to **Secret Server On-Premises**.

> For manual upgrades, please see the Secret Server On-Premises Upgrade Checklist prior to upgrading.

> If you have Privilege Manager installed, the Secret Server upgrade process will begin an upgrade for Privilege Manager as well.

How Standard Upgrades Work

Secret Server periodically polls the update server to detect new updates. If the "Allow Automatic Checks for Software Updates" option is enabled in the Admin > Configuration menu, you will see the "An update is available (xx.x.xxxxxx)" link after logging in with an administrator account.

Before You Begin

1. Ensure you will have access to account credentials for the server hosting Secret Server AND the SQL Server instance hosting your Secret Server database.
2. Ensure you have a recent backup of the application files and database available.
3. If you use clustering, stop the application pools on all of the servers except the one that is currently the "primary."

How to Upgrade

1. From a computer that has outbound network access, go to **Settings > All Settings** and click the **Upgrade Secret Server** link in the **Content** section. Admin > Upgrade. The Upgrade Secret Server page appears.
2. Ensure your install is backed up:

   > All your data is encrypted using the `encryption.config` file in your Secret Server application folder. **Your data cannot be decrypted without it**. Thus, it is critical that you backup the application folder and its contents before proceeding.

   a. Click the **Backup** button. The Admin > Backup tab appears. Here you can configure and then run a backup of the SQL Server database and the web application. You can save backups to local folders or network folders through configuration. The AppPool running Secret Server must be configured to not shut down. The

IIS Application pool will need Full Control to the backup folder specified for this user. Click the links on the tab to learn more before continuing.

b. Click the **Run backup now** button

c. When finished backing up both, return to the Upgrade Secret Server page and click to select the **The Secret Server database and application folder have been backed up** check box.

3. Click the **Continue** button. Another Upgrade Secret Server page appears.
4. Click the **Download Latest Version** button to download Secret Server. Wait for the download to finish. The Install Secret Server Upgrade page appears.
5. Click the **Upgrade** button. The upgrade starts. When it is finished, the Secret Server Upgrade Installation Status page appears.
6. Click the **Return to Home** button to return to the dashboard. The upgrade is complete.
7. If you intend to use Web clustering, proceed to "Upgrading Secret Server with Web Clustering" below.

Upgrading Secret Server with Web Clustering

This topic only applies to **Secret Server On-Premises**.

Introduction

Secret Server has a built-in Web installer. The Web installer is a series of pages inside Secret Server that allow you to download and run updates. Secret Server is accessible to users for most of the upgrade process. You can bring down outside access to the site if you want to prevent users from making changes during the upgrade. Preventing user access makes restoring the database and site backups simpler if you decide to roll back the upgrade immediately afterward.

You do not need to download the installer or `setup.exe`.

Please see the Secret Server On-Premises Upgrade Checklist prior to upgrading. Back up your Secret Server folder and database before performing the upgrade.

Never overwrite or delete your `encryption.config` file.

Distributed engines automatically upgrade.

Before Beginning

- Ensure that you have account credentials information and access for the server hosting Secret Server*and* the SQL Server instance hosting your Secret Server database.
- Have a recent backup of the application files and database available.
- If you use clustering, stop the application pools on all of the servers.

Upgrading a Clustered Environment

1. Follow the instructions in "Upgrading Secret Server" on page 125 or "Upgrading Secret Server Without Outbound Access" on page 149 as applicable to upgrade one server.
2. Once upgraded and working, copy the Web application folder (without the `database.config` or the `encryption.config` files) to all secondary servers, and replace the content of the existing Web application folder with the new version.
3. If the Delinea Management Server is installed and clustered, you need to copy the Delinea Management Server directory to the secondary servers as well. This directory is included by default for new installs of Secret Server 10.2 and above. The Delinea Management Server is used by advanced session recording and Privilege Manager. If the Delinea Management Server folder and site do not exist in IIS, then no additional actions are needed beyond copying the Secret Server directory.
4. Start secondary servers and confirm they still work.

> When copying the application files of the upgraded node, the log folder is optional.

EFS and DPAPI Encryption

When upgrading, after the initial cluster configuration, you do not need to copy the `database.config` or `encryption.config` files to the other servers. If you need to copy those files because the database configuration changed and uses DPAPI, disable DPAPI encryption in Secret Server by going to **Admin > Configuration** and clicking **Decrypt Key to not use DPAPI**, located on the **Security** tab, before copying those files to secondary servers.

> EFS encryption is tied to the user account running the Secret Server application pool, so it is not machine specific. Copying EFS encrypted files between Secret Server instances will not result in errors, but is not needed.

Upgrading Database Mirroring

1. If there is more than one Web server running Secret Server, ensure all instances are pointing to the same database.
2. Stop all but one of the web servers.
3. Perform the upgrade on that single instance.
4. Once upgraded and working, copy the Web application folder to all secondary servers.
5. Start the secondary servers, and confirm they work.
6. Ensure all instances are properly activated.
7. Ensure that the database changes have been replicated to the mirror database.
8. If the secondary Web server was pointing originally to the secondary database, adjust it to point back to the secondary database.

Upgrading Remote DR Instances

1. Perform the upgrade on one instance.
2. Backup that instance.
3. Copy the database backup to the remote DR instance.
4. Restore the database.
5. Once the instance is upgraded and working, copy the Web application folder (but not the `database.config` or the `encryption.config` files) to the remote DR instance (overwriting the existing files).
6. Restart IIS or recycle the application pool running Secret Server on the remote DR instance.
7. Confirm that the remote DR instance is working correctly.

Error Conditions

Error(s) that may arise if the following condition(s) exist:

The version does not match: If a node is not properly updated from the source node after an upgrade, that node will not run because the application version does not match the database. The solution is to copy the application folder (minus the `database.config` or `encryption.config` files) and to replace the files on the secondary server.

Minimizing Upgrade Downtime

This topic only applies to **Secret Server On-Premises**.

Introduction

Large enterprise Secret Server customers with clustered environments often have a strong interest in minimizing downtime during their upgrade process. This document details our recommendations for accomplishing that.

This strategy may require close coordination between networking, server administration, and SQL DBA teams. We recommend they are available at the same time during the upgrade to minimize downtime. This procedure works best for those upgrading from the prior most recent version of Secret Server.

We recommend that you have a QA or test environment mirroring your production environment and that you first run this procedure through that environment to ensure the desired results occur, prior to attempting this in a production environment. As with any third-party application upgrade, we strongly recommend taking snapshots or virtual machine backups of your web, Secret Server distributed engine, and database servers so you can easily revert to a pre-upgrade state if issue arise during the upgrade.

Distributed engines auto-upgrade as part of the upgrade process to the latest release. It is possible that you will have up to 10 minutes of downtime while the engines upgrade, regardless of this procedure.

Customers using IWA and upgrading from 10.6 or lower to 10.6 or greater need to follow the steps in "Configuring Integrated Windows Authentication" on page 383 to configure their distributed engines.

Procedures

Load Balanced Configuration Upgrade

Prerequisites

The upgrade procedure requires that you do these steps outside of the Secret Server install.

- Download the upgrade package (step 1)
- Backup your database (step 2)
- Obtain the database upgrade script from support (step 4)
- Backup any customized web.config or web-appsettings.config files (step 10)

Procedure

1. Download the latest version of Secret Server from the Support Website.
2. Perform a full backup of your Secret Server database using the preferred backup method used by your company. For a quicker recovery procedure in case of disaster, we recommend creating a local SQL backup, engaging your SQL DBA team as needed. If you use an AlwaysOn configuration, perform the backup from your primary node. If desired, you can choose the option to do a "copy only" backup to avoid interrupting any log truncation performed by your enterprise backup tool.
3. Restore the database onto a separate SQL server or separate instance within your environment. This restored backup is used to test the upgrade process. For this instruction, we call this server the "Test SQL Server."

   
   Depending on your circumstances, you might want to provision a standalone SQL server to accommodate this need in the future.

4. Request a database upgrade script from the Delinea Support team. You must provide them the current **exact** version of software you are on. The script can **only** be provided by Delinea. You can be request it prior to your upgrade.
5. Run the upgrade script on the Test SQL Server to verify the upgrade script runs without errors. If the upgrade script completes without errors, you can proceed.
6. Remove the database from the Test SQL Server. This may require your restarting SQL Server Engine services prior to removal.
7. Choose one of the web servers in a load balancer pool. For this instruction, we call this the "Target Web Server A." The pool has three servers total (Target Web Server A, B, and C).
8. Stop IIS on Target Web Server A. That server will appear offline in the pool list on your load balancer configuration application. For now, leave the other target web servers as is.
9. Temporarily remove Target Web Server A from the load balancer pool using your load balancer configuration application. Leave the server itself running. For example, on a F5 Big-IP load balancer:
   a. Click Local Traffic.
   b. Click Pools.

c. Click to select the desired pool.

d. Click Advanced in the Configuration section.

e. Ensure Reject is selected for Action on Service Down.

10. Copy the Secret Server application files you downloaded to Target Web Server A. Copy all files in the SS_update.zip file into Secret Server directory on Target Web Server A. This typically takes less than five minutes, depending on VM resources.

> If you have any customized settings that are per-node-specific in the existing web.config or web-appsettings.config files on the target servers, consider whether you want to protect those changes from being overwritten during the upgrade. Because the default contents of those files might change with the upgrade, we strongly recommend copying any customizations line-by-line to the new files, rather than simply replacing the new files with your customized ones. We suggest running a diff or comparison operation on the file pairs to see what, if anything, was customized in the existing files or changed in the new files.

11. When you are prompted, instruct the system copy dialog to overwrite all files.

12. (Optional) When the Secret Server files are finished copying, enable maintenance mode on Target Web Servers B and C to eliminate the possibility of password changes occurring during your database upgrade. The Secret Server read-only vaulting function is still possible from these nodes.

> See "Maintenance Mode FAQ" on page 217 for more on maintenance mode.

14. Start IIS on Target Web Server A. Wait until the site fully loads. This may take some time.

> Once the site loads, when you locally access that web server, you will see a Secret Server error message saying your database does not match your Secret Server. You can ignore that and click the Continue button.

15. Run the script you modified on the Production SQL Server.

16. Once the script completes, disable Target Web Servers B and C on your load balancer pool.

17. Enable Target Web Server A on your load balancer pool.

18. Access the web server through the load balancer URL. The Secret Server Login page should appear, and you should be able to log on immediately.

> If the warning message about the database not matching Secret Server appears, ignore it and click the Continue button.

19. Make Target Web Server A the primary load balancer node in the load balancer pool until the other target web servers are upgraded and online.

20. Disable IIS on Target Web Servers B and C.

21. Manually upgrade the Secret Server application files as previously discussed.

22. Enable IIS on Target Web Servers B and C.
23. Enable Target Web Servers B and C in the load balancer pool.
24. If you earlier put any of the target web servers in maintenance mode as a precaution, return them to normal function.

Manual Rolling Upgrade

Introduction

The manual rolling upgrade provides a way to upgrade Secret Server with little to no downtime. That is, users will continue to have secret access during the upgrade.

> This procedure only applies to clustered (multiple Web node) Secret Server environments environment.

Prerequisites

The administrator role needs the following permissions:

- Administer Configuration
- Administer Nodes
- Administer Backup

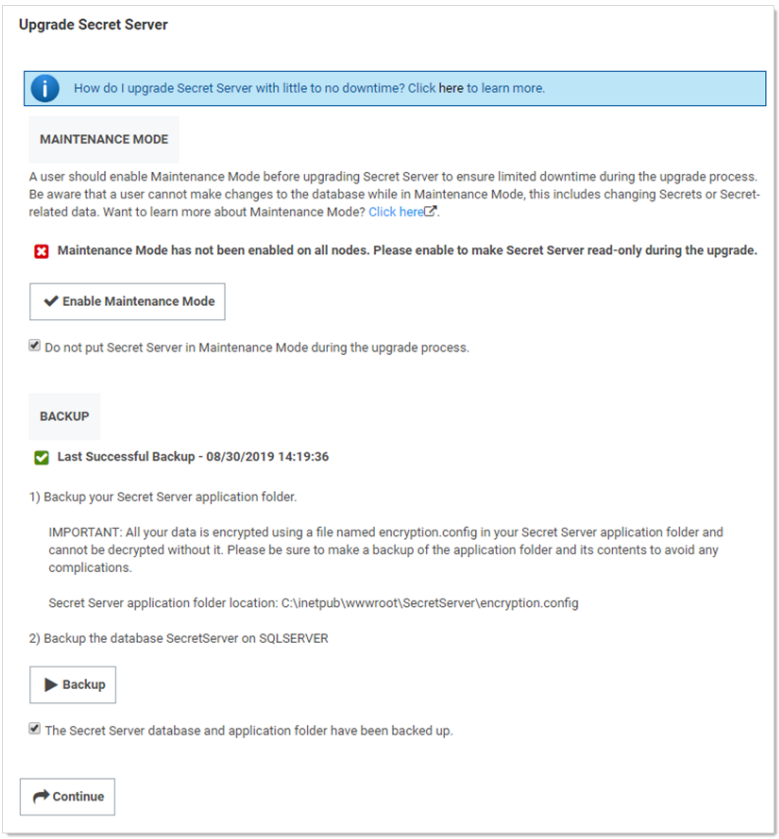
In addition, the role:

- Needs a database login with permission to change the database
- Requires access with permission to update files on web servers
- Must go through the current upgrade process
- Must not turn on maintenance mode until needed

Procedure

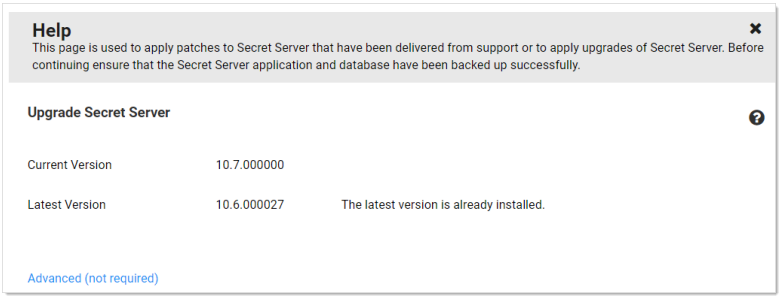
Task One: Uploading the Upgrade

1. Download the latest version of Secret Server.
2. Navigate to **Admin > See All > Upgrade Secret Server:**

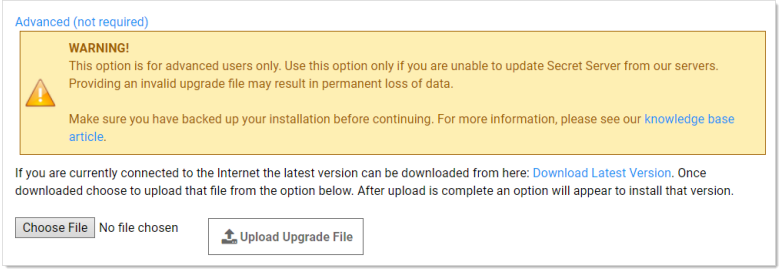


3. **Important:** Click to select the **Do not put Secret Server in Maintenance Mode during the upgrade process** check box.
4. Backup the Secret Server application folder.

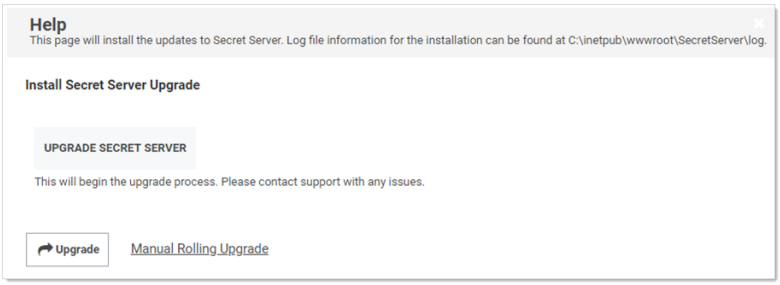
   Important: Ensure the encryption.config file is backed up. It is located at `c:\inetpub\wwwroot\SecretServer\encryption.config`.
5. Click the **Backup** button to back up the Secret Server database.
6. Click the **Continue** button. The Upgrade Secret Server page appears:



7. Click the **Advanced (not required)** link. The Advanced section appears:



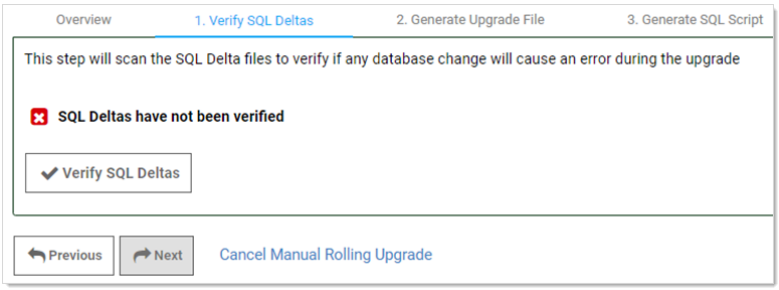
8. Click the **Choose File** button, and select the zip file you downloaded earlier to upgrade to.
9. Click the **Upload Upgrade File** button. The new version appears as available for installation:



10. Click the **Manual Rolling Upgrade** link. The Manual Rolling Upgrade wizard appears.

Task Two: Verifying SQL Changes (Wizard Step One)

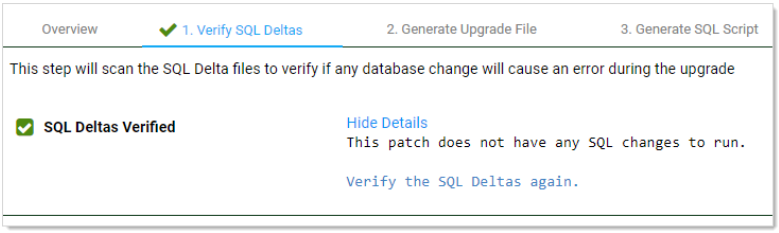
1. Click the **Next** Button. The Verify SQL Deltas tab appears:



> Clicking the "Cancel Manual Rolling Upgrade" link, at any time, will take you to the Install Secret Server Upgrade page.

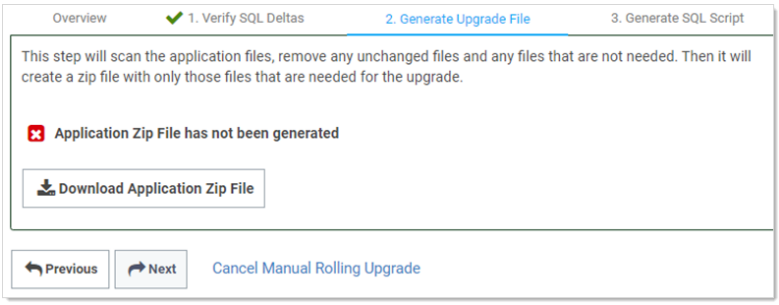
2. Click the **Verify SQL Deltas** button. This tests the prospective changes to see if errors result. If errors result,

please contact Delinea Technical Support. If the verification succeeds:



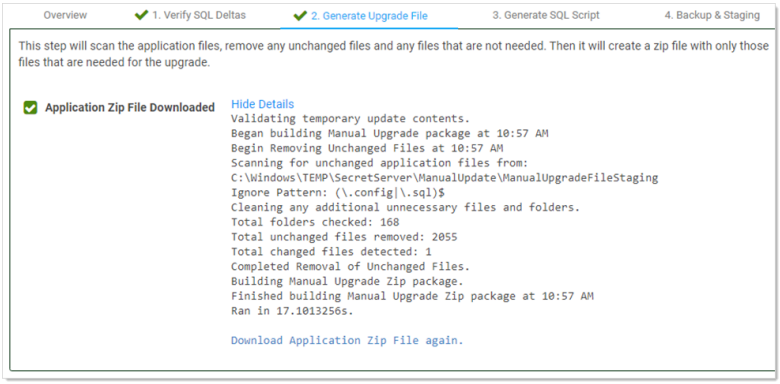
Task Three: Generating the Upgrade File (Wizard Step Two)

1. Click the **Next** button. The Generate Upgrade File tab appears:



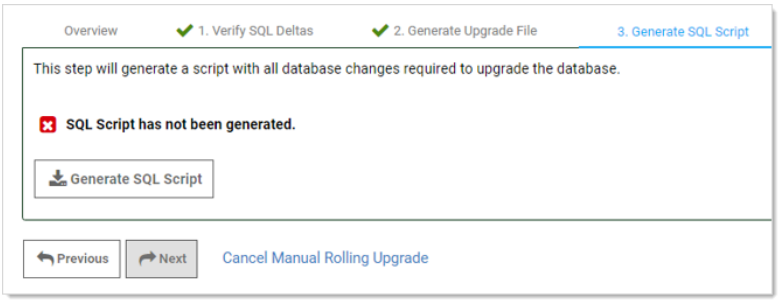
2. Click the **Download Application Zip File** button. This generates a zip file with only the changed files needed to upgrade the application files on the Web server nodes.

> This may take a few minutes to generate and download.

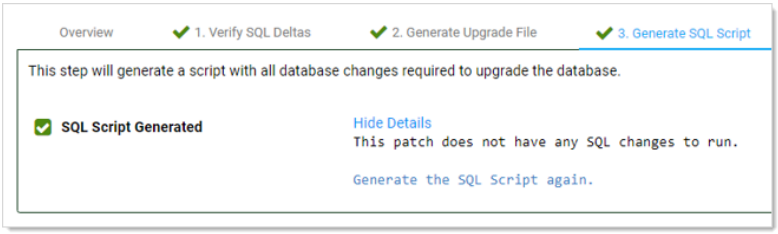


Task Four: Generating the SQL Script (Wizard Step Three)

1. Click the **Next** button. The Generate SQL Script tab appears:

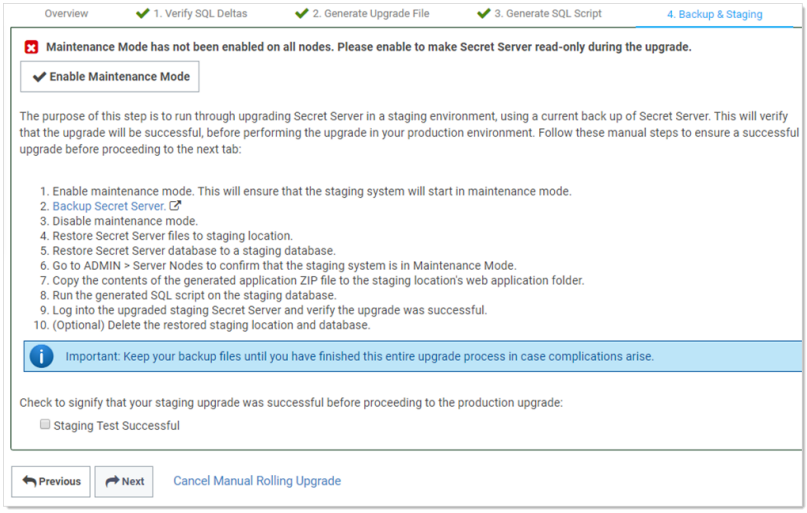


2. Click the **Generate SQL Script** button. This generates script file with all the database changes needed to upgrade the database. When finished:



   The wizard proceeds to step four:

Task Five: Backing up and Staging (Wizard Step Four)



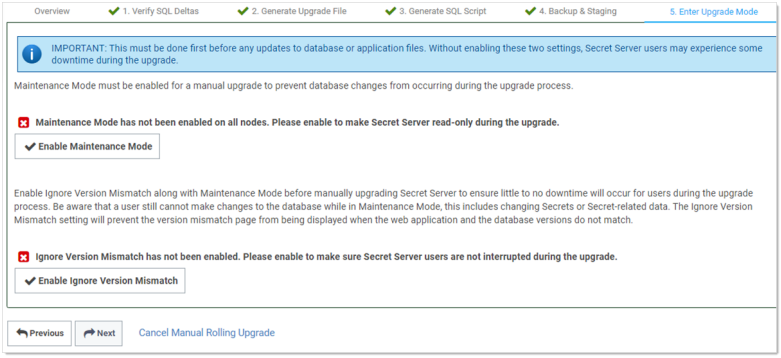
1. Click the **Enable Maintenance Mode** button.
2. Back up Secret Server: Type "backup" in the Admin search text box, and click the item that appears in the dropdown list to access the Backup Configuration page. Click the **Backup Now** button.

3. Click the **Disable Maintenance Mode** button.
4. Restore Secret Server files to the staging location:
   a. Copy the backup zip file to the staging location.
   b. Unzip the backup file.
   c. Copy the files to the web application folder.
5. Restore the Secret Server database to a staging database:
   a. In SQL Server Management Studio, right click on **Databases**.
   b. Click **Restore Database**.
   c. In **Source**, select **Device**.
   d. Select and add the backup database file location.
   e. Click **Ok**.
6. Go to **Admin > Secret Nodes** to confirm the staging system is in maintenance mode.
7. Copy the contents of the generated application Zip file to the staging location's web application folder. Typically, this is `C:\inetpub\wwwroot\SecretServer`.
8. Run the generated SQL script on the staging database.
9. Log on the upgraded staging Secret Server to verify the upgrade was successful.
10. (Optional) Delete the restored staging location and database.

    Important: Keep the backup files till you verify the upgrade was successful. You may need them if an issue develops.
11. Click to select the **Staging Test Successful** check box to confirm your staging upgrade was successful. This is your confirmation that there were no errors before performing the actual upgrade in your production environment. The confirmation is recorded.

Task Six: Starting Upgrade Mode (Wizard Step Five)

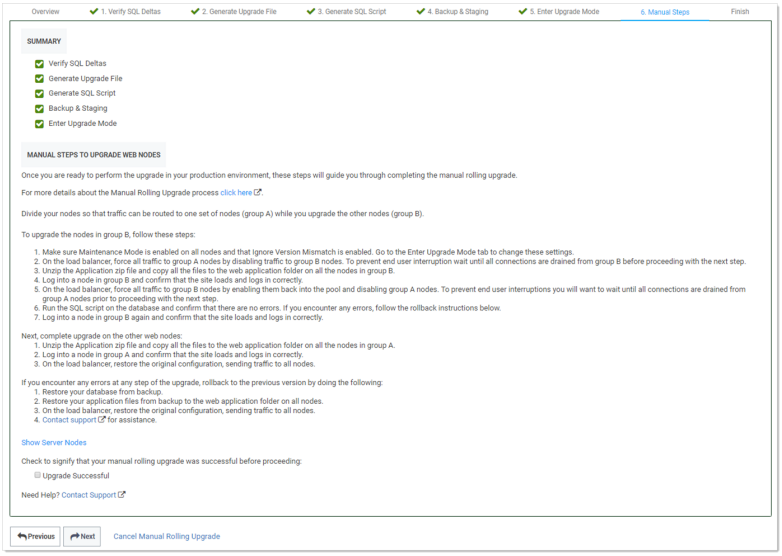
1. Click the **Next** button. The Enter Upgrade Mode tab appears:



2. Click the **Enable Maintenance Mode** button. This mode limits the activities of users on secrets, secret templates, password requirements, and others and can take several minutes to start. A confirmation popup

appears.

3. Click the **Enable** button to confirm the mode change. The popup disappears.
4. Click the **Enable Ignore Version Mismatch** button. This prevents users from being redirected to the Version Mismatch page. A confirmation popup appears.
5. Click the **Enable** button to confirm the setting change. The popup disappears.
6. Click the **Next** button. The Manual Steps tab appears:



Task Seven: Upgrading Web Nodes (Wizard Step Six)

To upgrade Web nodes:

1. Split your nodes into two approximately even groups (A and B) so that one group can service traffic while the other is upgrading.
2. Ensure "maintenance mode" and "ignore version mismatch" are enabled on each node. You can change them from the Enter Upgrade Mode tab.
3. On the load balancer, disable traffic to group B. To prevent traffic interruptions, ensure those nodes are all completely disabled before proceeding to the next step. Group A, alone, now handles the traffic. For example, on a F5 Big-IP load balancer you:
   a. Select the Members tab on the pool page.
   b. Select the node to disable.
   c. Click Force Offline.
4. For each node in group B:
   a. Navigate to the Downloads folder.
   b. Extract all the files from the application zip file downloaded earlier.

c. Copy the extracted files to the Web application folder.

d. Log onto the node to ensure the site correctly loads and logs on.

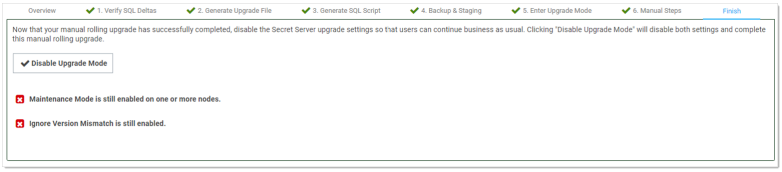
5. On the load balancer, enable the group B nodes to return them to the pool.
6. Disable traffic to group A. To prevent traffic interruptions, ensure those nodes are all completely disabled before proceeding to the next step. Group B, alone, now handles the traffic.
7. Execute the script you created on the database, confirming there are no errors. If there are errors, follow the rollback instructions.
8. Log onto each group B node again to ensure the site correctly loads and logs on.
9. For each node in group A:

   a. Navigate to the Downloads folder.

   b. Extract all the files from the application zip file downloaded earlier.

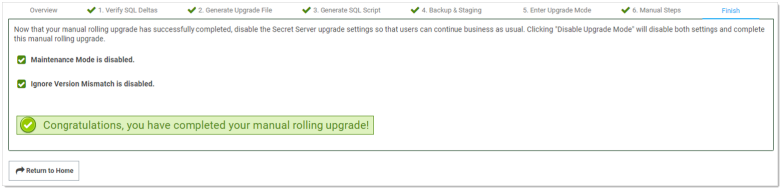
   c. Copy the extracted files to the Web application folder.

   d. Log onto the node to ensure the site correctly loads and logs on.
10. On the load balancer, enable the group A nodes to return them to the pool, restoring the original configuration and returning traffic to all nodes.
11. Click to select the **Upgrade Successful** check box.
12. Click the **Next** button. The Finish tab appears:



Task Eight: Finishing up (Wizard Step Seven)

1. Click the **Disable Upgrade Mode** button. A Finish Manual Rolling Upgrade popup appears. This popup both disables maintenance mode and disables the ignore version mismatch setting.
2. Click the **Disable** button. The popup disappears, and a completion message appears:



Troubleshooting and Notes

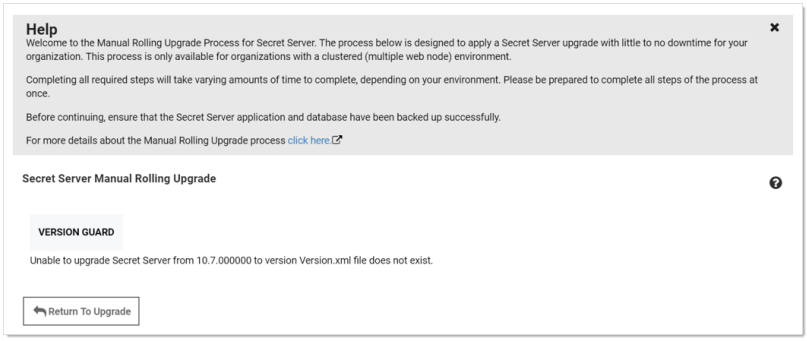
Rolling Back to the Previous Version

If you encounter errors at any step of the upgrade, rollback to the previous Secret Server version:

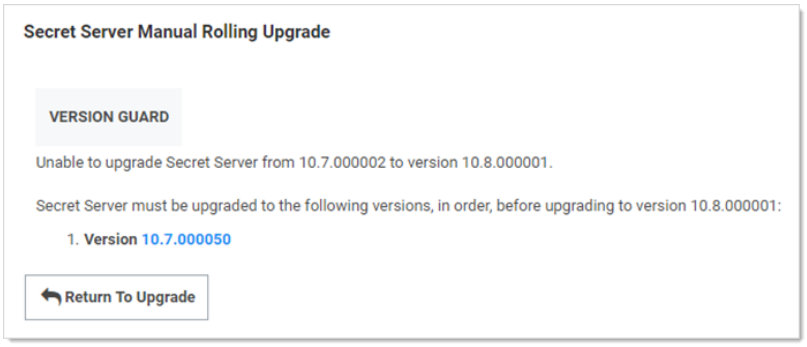
1. Restore the database from the backup.
2. Restore the application files from the backup files to the Web application folder on all nodes.
3. On the load balancer, restore the original configuration, sending traffic to all nodes.
4. For assistance, contact us.

Version Guard

If an uploaded upgrade file cannot be used to upgrade the current version of Secret Server, then "Version Guard" will block the upgrade and provide instructions on how to continue:



This usually occurs when not completing the perquisite steps in order. Click the **Return To Upgrade** button to return you to the first upgrade page to remedy the situation.



This page also list the blocking versions that you must upgrade to prior to running the manual rolling upgrade.

New Advanced Configuration Setting

There is a new setting called "Manual Upgrade: Allow version mismatch while in Maintenance Mode." This setting, which only applies in maintenance mode, prevents Secret Server from redirecting users to the version mismatch message page.

New Audit Type

To support the manual rolling upgrade, there is a new audit type–ManualUpgrade. Its audits are stored in the tbAudit table and record the following actions:

- CANCEL
- COMPLETED
- GENERATE DB SCRIPT
- GENERATE UPGRADE ZIP
- STAGING TEST
- STARTED
- VERIFY DELTAS

Manual Rolling Upgrade

This topic only applies to **Secret Server On-Premises**.

Introduction

The manual rolling upgrade provides a way to upgrade Secret Server with little to no downtime. That is, users will continue to have secret access during the upgrade.

This procedure only applies to clustered (multiple Web node) Secret Server environments environment.

Prerequisites

The administrator role needs the following permissions:

- Administer Configuration
- Administer Nodes
- Administer Backup

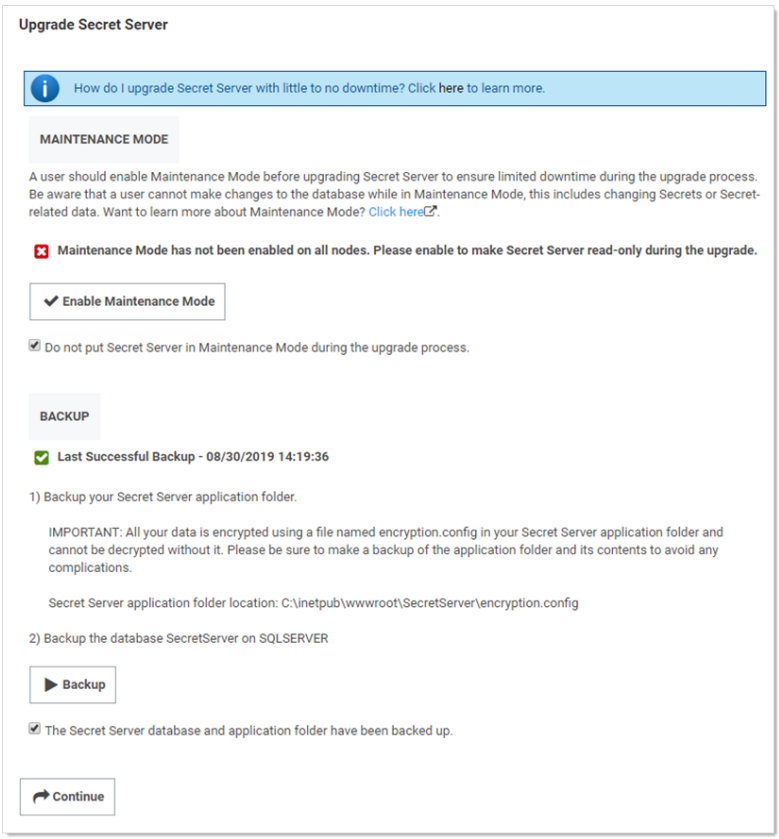
In addition, the role:

- Needs a database login with permission to change the database
- Requires access with permission to update files on web servers
- Must go through the current upgrade process
- Must not turn on maintenance mode until needed

Procedure

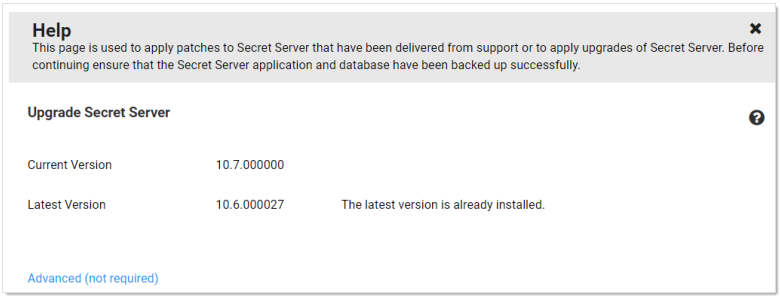
Task One: Uploading the Upgrade

1. Download the latest version of Secret Server.
2. Navigate to **Admin > See All > Upgrade Secret Server:**

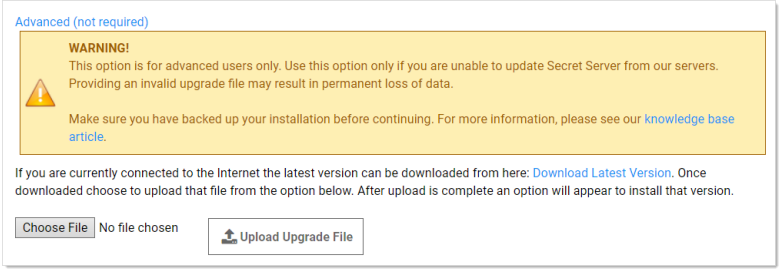


3. **Important:** Click to select the **Do not put Secret Server in Maintenance Mode during the upgrade process** check box.
4. Backup the Secret Server application folder.

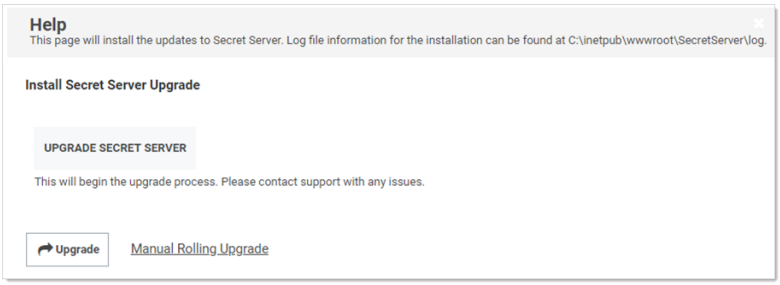
   Important: Ensure the encryption.config file is backed up. It is located at `c:\inetpub\wwwroot\SecretServer\encryption.config`.
5. Click the **Backup** button to back up the Secret Server database.
6. Click the **Continue** button. The Upgrade Secret Server page appears:



7. Click the **Advanced (not required)** link. The Advanced section appears:



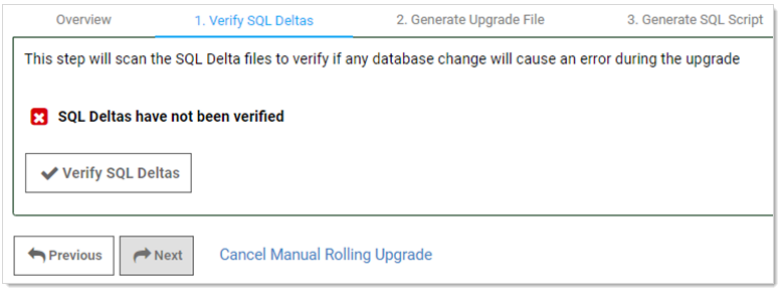
8. Click the **Choose File** button, and select the zip file you downloaded earlier to upgrade to.
9. Click the **Upload Upgrade File** button. The new version appears as available for installation:



10. Click the **Manual Rolling Upgrade** link. The Manual Rolling Upgrade wizard appears.

Task Two: Verifying SQL Changes (Wizard Step One)

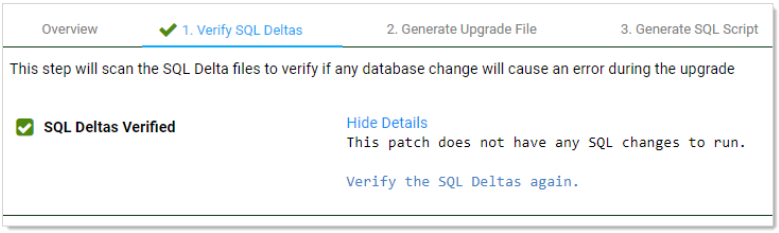
1. Click the **Next** Button. The Verify SQL Deltas tab appears:



> Clicking the "Cancel Manual Rolling Upgrade" link, at any time, will take you to the Install Secret Server Upgrade page.

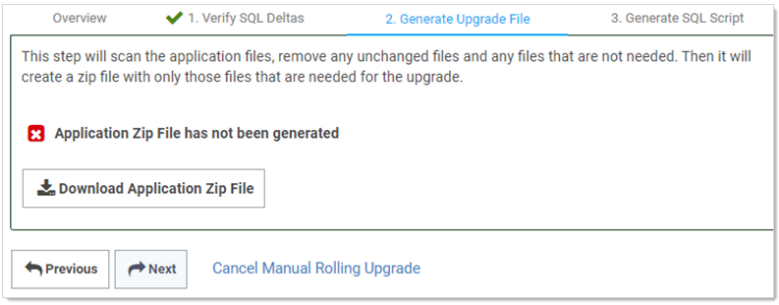
2. Click the **Verify SQL Deltas** button. This tests the prospective changes to see if errors result. If errors result,

please contact Delinea Technical Support. If the verification succeeds:



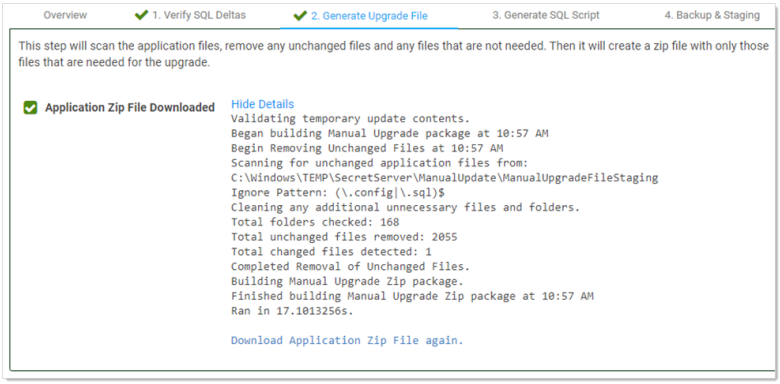
Task Three: Generating the Upgrade File (Wizard Step Two)

1. Click the **Next** button. The Generate Upgrade File tab appears:



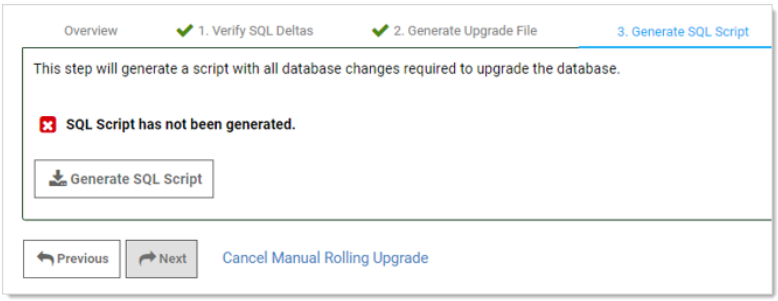
2. Click the **Download Application Zip File** button. This generates a zip file with only the changed files needed to upgrade the application files on the Web server nodes.

> This may take a few minutes to generate and download.

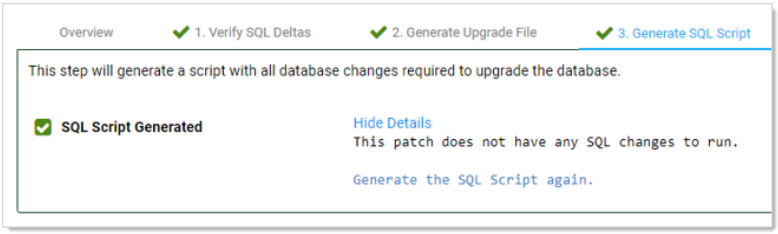


Task Four: Generating the SQL Script (Wizard Step Three)

1. Click the **Next** button. The Generate SQL Script tab appears:

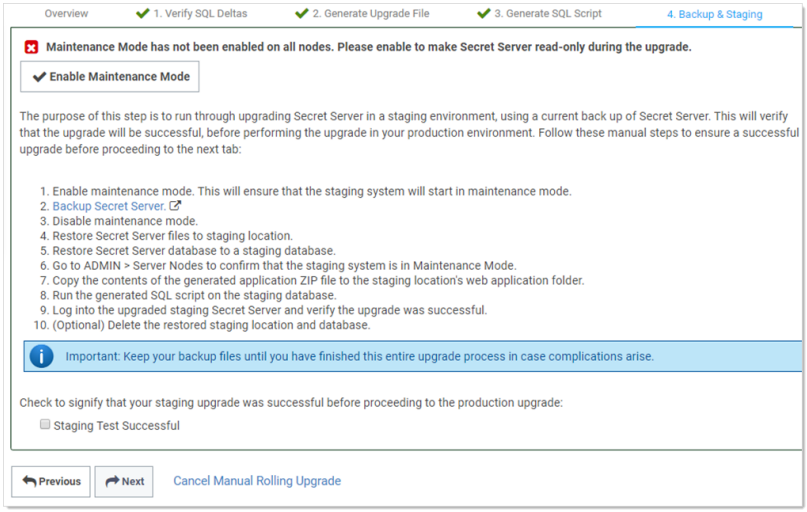


2. Click the **Generate SQL Script** button. This generates script file with all the database changes needed to upgrade the database. When finished:



The wizard proceeds to step four:

Task Five: Backing up and Staging (Wizard Step Four)



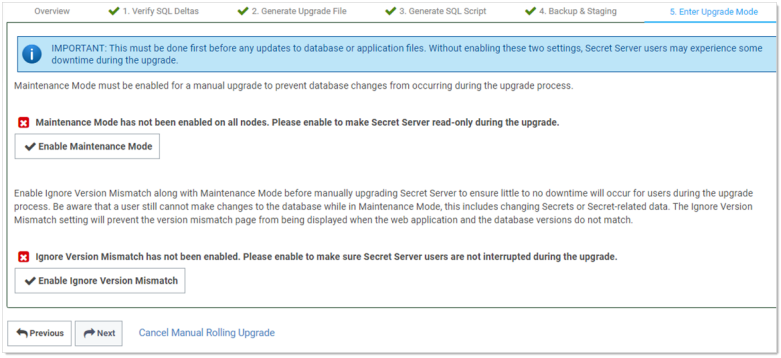
1. Click the **Enable Maintenance Mode** button.
2. Back up Secret Server: Type "backup" in the Admin search text box, and click the item that appears in the dropdown list to access the Backup Configuration page. Click the **Backup Now** button.

3. Click the **Disable Maintenance Mode** button.
4. Restore Secret Server files to the staging location:
   a. Copy the backup zip file to the staging location.
   b. Unzip the backup file.
   c. Copy the files to the web application folder.
5. Restore the Secret Server database to a staging database:
   a. In SQL Server Management Studio, right click on **Databases**.
   b. Click **Restore Database**.
   c. In **Source**, select **Device**.
   d. Select and add the backup database file location.
   e. Click **Ok**.
6. Go to **Admin > Secret Nodes** to confirm the staging system is in maintenance mode.
7. Copy the contents of the generated application Zip file to the staging location's web application folder. Typically, this is `C:\inetpub\wwwroot\SecretServer`.
8. Run the generated SQL script on the staging database.
9. Log on the upgraded staging Secret Server to verify the upgrade was successful.
10. (Optional) Delete the restored staging location and database.

    Important: Keep the backup files till you verify the upgrade was successful. You may need them if an issue develops.
11. Click to select the **Staging Test Successful** check box to confirm your staging upgrade was successful. This is your confirmation that there were no errors before performing the actual upgrade in your production environment. The confirmation is recorded.

Task Six: Starting Upgrade Mode (Wizard Step Five)

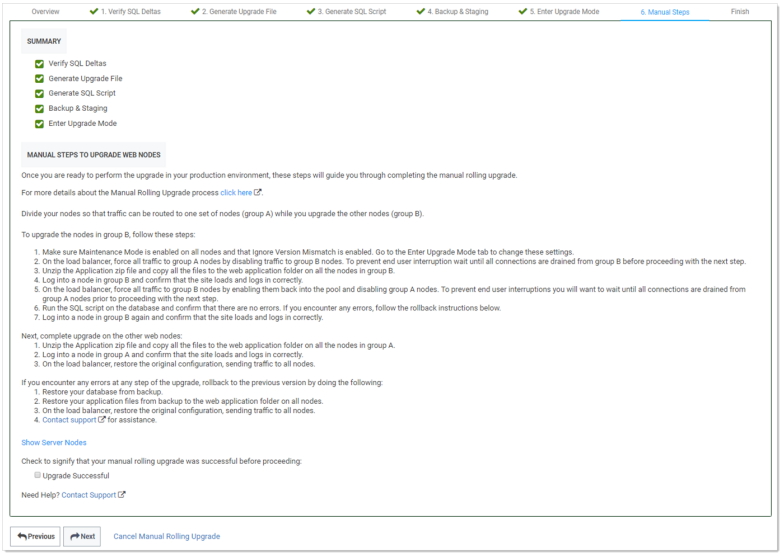
1. Click the **Next** button. The Enter Upgrade Mode tab appears:



2. Click the **Enable Maintenance Mode** button. This mode limits the activities of users on secrets, secret templates, password requirements, and others and can take several minutes to start. A confirmation popup

appears.

3. Click the **Enable** button to confirm the mode change. The popup disappears.
4. Click the **Enable Ignore Version Mismatch** button. This prevents users from being redirected to the Version Mismatch page. A confirmation popup appears.
5. Click the **Enable** button to confirm the setting change. The popup disappears.
6. Click the **Next** button. The Manual Steps tab appears:



Task Seven: Upgrading Web Nodes (Wizard Step Six)

To upgrade Web nodes:

1. Split your nodes into two approximately even groups (A and B) so that one group can service traffic while the other is upgrading.
2. Ensure "maintenance mode" and "ignore version mismatch" are enabled on each node. You can change them from the Enter Upgrade Mode tab.
3. On the load balancer, disable traffic to group B. To prevent traffic interruptions, ensure those nodes are all completely disabled before proceeding to the next step. Group A, alone, now handles the traffic. For example, on a F5 Big-IP load balancer you:
   a. Select the Members tab on the pool page.
   b. Select the node to disable.
   c. Click Force Offline.
4. For each node in group B:
   a. Navigate to the Downloads folder.
   b. Extract all the files from the application zip file downloaded earlier.

c. Copy the extracted files to the Web application folder.

d. Log onto the node to ensure the site correctly loads and logs on.

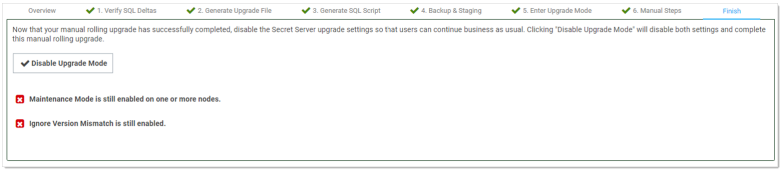
5. On the load balancer, enable the group B nodes to return them to the pool.
6. Disable traffic to group A. To prevent traffic interruptions, ensure those nodes are all completely disabled before proceeding to the next step. Group B, alone, now handles the traffic.
7. Execute the script you created on the database, confirming there are no errors. If there are errors, follow the rollback instructions.
8. Log onto each group B node again to ensure the site correctly loads and logs on.
9. For each node in group A:

   a. Navigate to the Downloads folder.

   b. Extract all the files from the application zip file downloaded earlier.

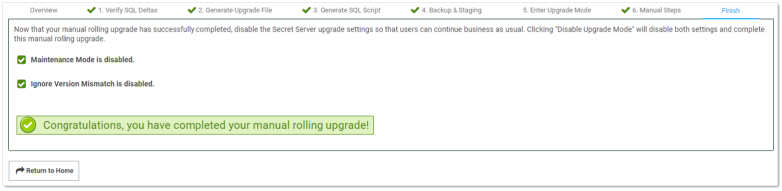
   c. Copy the extracted files to the Web application folder.

   d. Log onto the node to ensure the site correctly loads and logs on.
10. On the load balancer, enable the group A nodes to return them to the pool, restoring the original configuration and returning traffic to all nodes.
11. Click to select the **Upgrade Successful** check box.
12. Click the **Next** button. The Finish tab appears:



Task Eight: Finishing up (Wizard Step Seven)

1. Click the **Disable Upgrade Mode** button. A Finish Manual Rolling Upgrade popup appears. This popup both disables maintenance mode and disables the ignore version mismatch setting.
2. Click the **Disable** button. The popup disappears, and a completion message appears:



Troubleshooting and Notes

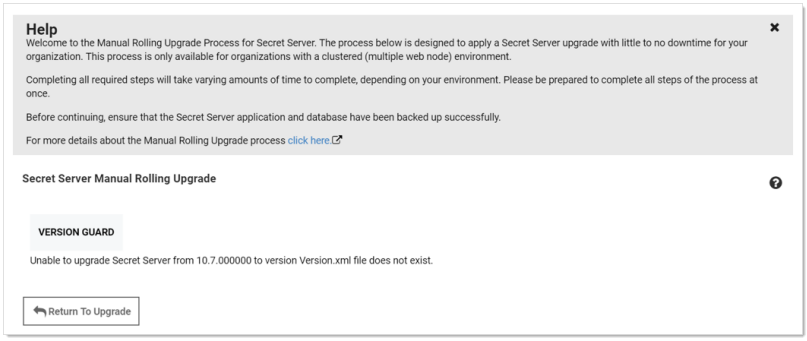
Rolling Back to the Previous Version

If you encounter errors at any step of the upgrade, rollback to the previous Secret Server version:

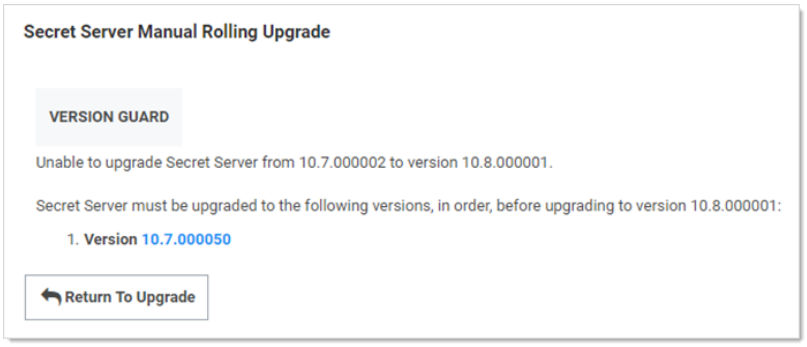
1. Restore the database from the backup.
2. Restore the application files from the backup files to the Web application folder on all nodes.
3. On the load balancer, restore the original configuration, sending traffic to all nodes.
4. For assistance, contact us.

Version Guard

If an uploaded upgrade file cannot be used to upgrade the current version of Secret Server, then "Version Guard" will block the upgrade and provide instructions on how to continue:



This usually occurs when not completing the perquisite steps in order. Click the **Return To Upgrade** button to return you to the first upgrade page to remedy the situation.



This page also list the blocking versions that you must upgrade to prior to running the manual rolling upgrade.

New Advanced Configuration Setting

There is a new setting called "Manual Upgrade: Allow version mismatch while in Maintenance Mode." This setting, which only applies in maintenance mode, prevents Secret Server from redirecting users to the version mismatch message page.

New Audit Type

To support the manual rolling upgrade, there is a new audit type–ManualUpgrade. Its audits are stored in the tbAudit table and record the following actions:

- CANCEL
- COMPLETED
- GENERATE DB SCRIPT
- GENERATE UPGRADE ZIP
- STAGING TEST
- STARTED
- VERIFY DELTAS

Upgrading Secret Server Without Outbound Access

This topic only applies to **Secret Server On-Premises**.

Upgrading to Secret Server version 8.9.000000 and above will require **Windows Server 2008 R2 or greater.**

Upgrading to Secret Server version 8.5.000000 and above, there are changes in the .NET Framework version you will need to be aware of along with some additional steps in the upgrade process. For more information, see Upgrading Secret Server Without Outbound Access.

Upgrading to Secret Server version 10.0.000000 and above will require configuring integrated pipeline mode on the Secret Server Application Pool. Please see "Changing IIS to Not Stop Worker Process in IIS 7.0 and Later" on page 238 for details on configuring integrated pipeline mode in IIS. If using Integrated Windows Authentication you will also need to update IIS authentication settings as detailed in "Configuring Integrated Windows Authentication" on page 383. If you are at version 9.1.000000 and below, you will need to first upgrade to 9.1.000001 before you can upgrade to 10.0.000000 and above.

How Upgrades Work

Secret Server periodically polls our update server to detect updates. If your Secret Server is on an internal network that has no outbound access or goes through a proxy, Secret Server will not be able to perform updates automatically, therefore, outbound access to the below connections on your firewall is needed if you want to perform updates automatically:

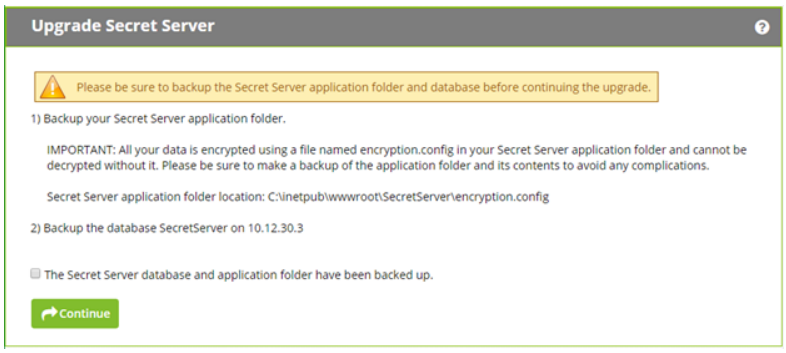
- `d36zgw9sidnotm.cloudfront.net:443`
- `updates.thycotic.net:443`
- `updates.thycotic.net:80`

The steps below can be used to perform an upgrade for versions 7.1.000015 and higher. If you have an older version of Secret Server, please contact Delinea technical support for assistance.

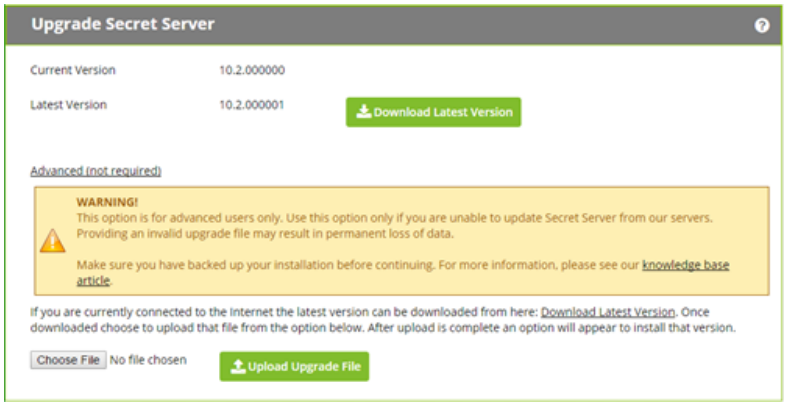
Procedure

Step 1: Open the Upgrade Secret Server Wizard

1. From a computer that does have outbound network access and Secret Server access, go to the Secret Server Upgrade page by browsing to: `http://<yourinstance>/Installer.aspx?patch=true` (filling in your Secret Server URL for `<yourinstance>`). The wizard appears:



2. Backup your Secret Server application folder and your Secret Server database.
3. Click to select the **Secret Server database...** check box on the page.
4. Click the **Continue** button. The next page appears:



Step 2: Get and Upload the Latest .zip File

1. Download the latest version .zip file by clicking the **Download Latest Version** button on the installer page. The file name will appear something like `Version_10_2_000000.zip`. Note where you save it.

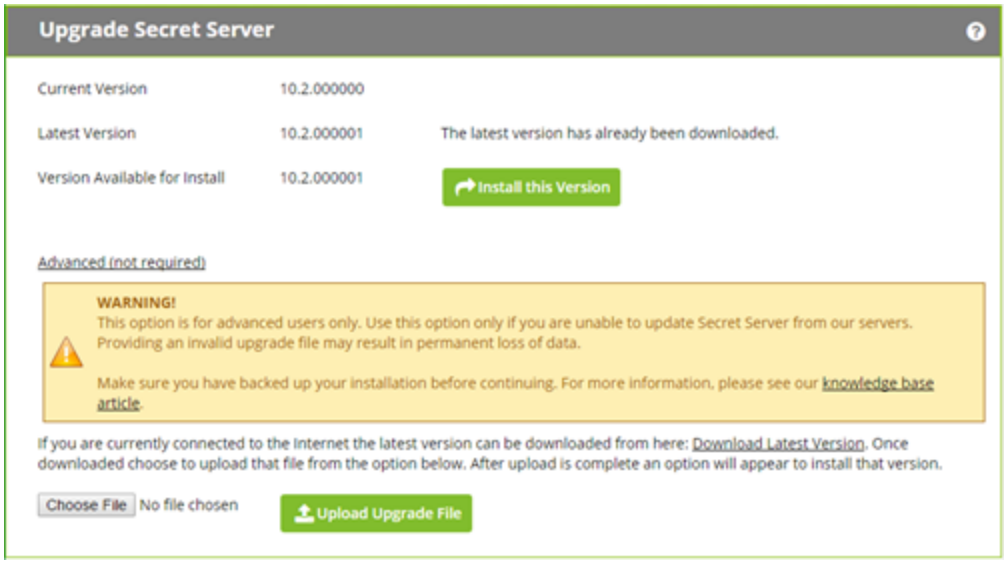
> You also can find the downloadable update files below.

2. Click the **Choose File** button to select the Secret Server .zip file you just downloaded.

> To verify the file hashes for the latest version using the posted hash values, refer to "Secret Server Download Hashes" on page 47.

> You should **not** use the fresh install SecretServer.zip or setup.exe that is first downloaded from Delinea.com. Only use the Get Latest Version link–there is a difference between the upgrade file and fresh install zip.

3. Click the **Upload Upgrade File** button. You see a message confirming the file was successfully uploaded, and the Install This Version button appears.



4. Click the **Install this Version** button. The Upgrade Secret Server page appears (not shown).

Step 3: Upgrade Secret Server

1. Click the **Upgrade** button. The upgrade automatically processes and once it has finished you will see a confirmation page.
2. Click **Return to Home** to return to the dashboard.

Offline Installation Download Files

If you do not have access to another installation of Secret Server or you are upgrading from an earlier version, click one of the following links, depending on your *current* installed version:

- 8.4.000003 or earlier
- 8.4.000004 to 9.1.000000
- 9.1.000001 to 10.9.000003
- 10.9.000005 to 11.5.000002
- 11.5.000002 or later

Navigation and Customization

Customization

Navigation

Customization

"Setting Color Modes" on page 155

"Configuring Custom Logos" below

"Configuring Global Banners" on the next page

"Enabling Login Banners" on page 154

Configuring Custom Logos

You can add your own branding by adding customized logos to Secret Server. The logos only apply to the current UI—the classic UI is not supported. The logos and their requirements are listed in the table below.

Table: Custom Logo Specifications

Mode	Format	Required Pixel Dimensions	Maximum File Size
Light	Full-sized	200×50	1 MB
Light	Icon	50×50	1 MB
Dark	Full-sized	200×50	1 MB
Dark	Icon	50×50	1 MB
Light	Login	400×400	1 MB
Dark	Login	400×400	1 MB

To use custom logos:

1. Create images for all six of the listings in the table above. The images can be svg, png, or jpg. The images must be the exact dimensions listed and not over the maximum file size.
2. Go to **Admin > User Interface**.
3. Click the **Edit** button. The page becomes editable.
4. For each of the listed custom logos:
   a. Click the **Change** link. A file selection dialog box appears.
   b. Navigate to and select the image matching the description.
   c. Hover over the logo description on the left to see the exact size requirements for the specific logo.
5. Click **Save**. The images appear in the listing, and the logos are deployed to the application at the top of the user's profile.

Configuring Global Banners

Overview

You can configure a multipurpose global banner for all users and use it for maintenance, security, or policy notifications. You can set the banner style, text, a hyperlink, and an in-theme color, which is determined by the style. The styles include:

- Caution
- Error
- Failure
- Information
- Status
- Success
- Warning

The settings are available in the User Interface section.

Configuration

1. Navigate to **Admin > User Interface.**
2. Click **Edit**. The page becomes editable.
3. Click to select the **Show Custom Banner** check box. New controls appear below.
4. Type or paste the desired text in the **Banner Text** text box.
5. Type or paste the desired URL, if any, in the **Banner View Link** text box.
6. Click the **Banner Style** dropdown list to select one of the following:
   - Caution
   - Error
   - Failure
   - Information
   - Status
   - Success
   - Warning
7. Click **Save**.
8. Click **Preview Banner** to see your new banner.

   The Preview button is primarily for when you have previously dismissed the banner. The first time you save a new banner, it automatically appears at the top of every page, including the one you are on.

9. Click the **View** button inside your banner to go to the URL you entered.
10. Click the **Dismiss** button to remove the banner from all pages for that session.

Enabling Login Banners

If your company requires the login banner for usage agreements and conditions to be visible when users log into Secret Server:

To enable the login banner, follow the procedure below:

1. Go to **Admin > Login policy** and click **Edit** at the top right.
2. Select the checkbox for **Enable Login Policy**. The Force login policy check box and Login policy window will appear.

   Check **Force login policy** to force the user to accept the login policy that is displayed.

   Edit the text in the **Login policy** window if needed and click **Save.**
3. Log out of Secret Server and re-try logging in.

   On the Secret Server Login page, users logging into Secret Server will see the default message provided in the login policy box.

Note that enabling login policy will override the value in "policy.txt", and your changes will not be lost after the upgrades.

The login policy is not applicable to the mobile app and will not be shown.

Setting Color Modes

Overview

Important: As of version 11.4, the classic UI is no longer available to anybody.

Color mode is a color scheme (skin) for the UI. The color modes are System Default, Light, and Dark. System default means whatever color was chosen on your system for the Windows application default.

Setting Your Default Color Mode

1. Click the user icon at the top right of any page and select **User Preferences**. The User Preferences Page appears.
2. Click the **Settings** tab.
3. Click the **Color Mode** dropdown list to select the desire color mode. Your choices are:
   - Light Mode
   - Dark Mode
   - System Default: The color chosen for your Windows default application color.

   The mode changes right away—no need to save the change.

Navigation

"All Secrets Page" on page 1117

"Application Dashboard" on page 157

"Configuration Search" on page 161

"Global Search" on page 161

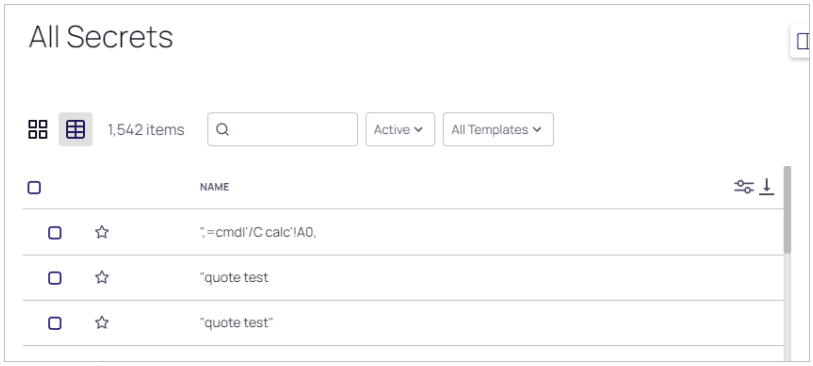
"Main Navigation Drawers" on page 162

"Secret Folder-Tree Panel" on page 1072

All Secrets Page

All Secrets is a master table of the secrets stored on Secret Server. It is a one-stop, searchable location for examining the status and properties of secrets. It is a supplement to, not a replacement for, the "Folders" on

page 1059. It lists and you can sort by secret template, heartbeat status, sync status, machine, access date, username, and much more.



Click the **Secrets** menu item in the left menu to see the All Secrets Page. Click the **>>** icon to see the All Secrets folder tree.

Secret Columns

You can customize which columns are displayed by clicking the on the right side of the title bar. The sortable columns available are (the ones displayed by default are bolded):

- Auto Change Enabled
- Checked Out
- Checkout Enabled
- Created
- Days until Expiration
- Deleted
- DoubleLock Enabled
- **Folder**
- **Heartbeat**
- Hide Password
- Inherits Permissions
- **Last Accessed (When the logged in user last accessed a secret, not the last time a secret was accessed by anyone)**
- Machine
- **Name**
- Notes
- **Out of Sync**
- Requires Approval

- Requires Comment
- **Secret Template**
- Username

Quick Launch and Actions

You can quickly access several secret actions from the All Secrets table by hovering the mouse pointer over the secret's row:

- Quick Launch: Secrets that display a rocket icon have secret launchers associated with them. Click the icon and a Select Launcher page appears. Click on the desired launcher from the list.
- More Actions: Click the **...** icon to
  - Audit: Brings you to the Audit tab of the secret's page.
  - Details: Brings you to the General tab of the secret's page.
  - Enter Comment: Brings you to a page for entering a comment before you are allowed to view the secret. Comment ask for information such as reason for viewing and ticketing system.
  - Share: Brings you to the Sharing tab of the secret's page.

Other Features

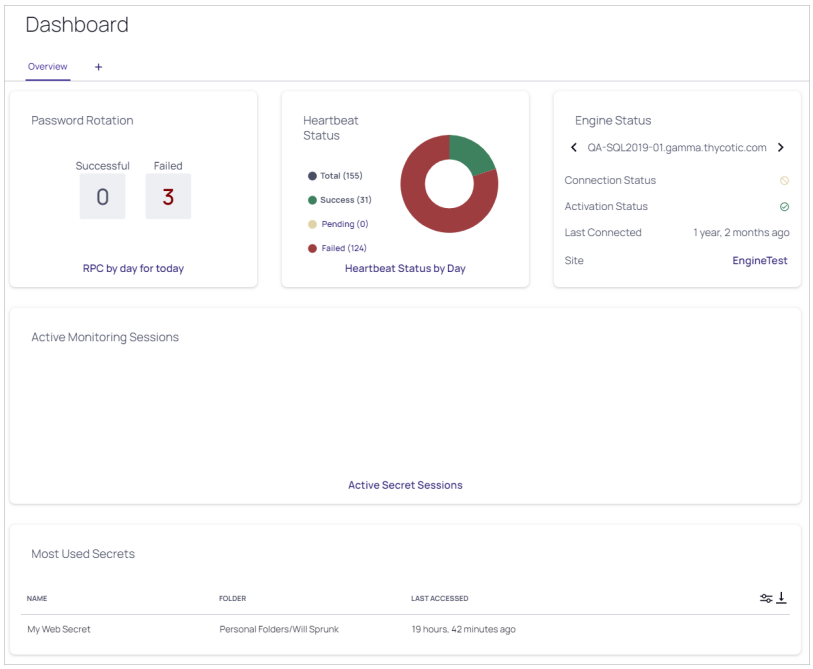
The All Secrets page also allows you to:

- Select multiple secrets for "Running Dashboard Bulk Actions" on page 160 by clicking the secret's check box. You can also download multiple secrets in a CSV file by clicking the download icon at the top of the table.
- Select the secret as a favorite by clicking the star icon.
- View supplemental information about the secret by clicking any blank area on the secret's row.
- Click the secret's name to go to that secret's page.
- Filter the secrets shown in the table by template, search text, or activity status.
- Click the "Secret Navigation Slideout" on page 1153.

Application Dashboard

The Secret Server dashboard is the main page for searching and viewing secrets. You access it by clicking the Dashboard menu item.

Figure: Secret Server Dashboard



Dashboard Components

Secret Panel

Click the **Open Secret Panel** button in the top right to view the Secret panel. The panel contains:

- **Favorites:** A list of your favorite secrets, which you manually tagged by clicking the star icon on the All Secrets page.
- **Recent:** A list of your recently used secrets.
- **Most Used Secrets:** A list of your most frequently used secrets.
- **Shared with Me:** A list of secrets others shared with you.

Overview Tab

The Overview tab provides several fixed widgets for getting a quick understanding of your Secret Server installation:

- **Approvals:** Your current in-process approvals. See "Access Request Overview" on page 1074.
- **Engine Status:** A scrollable list of distributed engine connections and activation statuses.
- **Heartbeat Status:** A graphic of the current status of your heartbeats: success, pending, or failed. When you click on one of the statuses, you are brought to a report page for that status. For example, **Reports > Secrets Failing Heartbeat**. When you click the **Heartbeat Status by Day** link , you are brought to the **Reports > Heartbeat Status by Day** page. See "Heartbeat Overview" on page 1042.
- **Most Used Secrets:** A table of the most recently accessed secrets, listed by date and folder.

- **Password Rotation:** The state of your current password rotations. When you click the **RPC by Day for today** link you are brought to the **Reports > RPC by Day** report page. See "RPC Overview" on page 904.

> To see an overview of incoming system and subscription alerts, see the "Notification Inbox Overview" on page 319.

Customized Tabs

Tab Management

The following operations are available for creating custom tabs:

- **Create:** Click the **+** icon to the right of the tabs to create a new empty tab.
- **Delete:** Click **Remove tab page** on a tab and select **Confirm remove tab** to delete a tab. You can cancel changes by clicking **Cancel**. A confirmation pop up page appears.
- **Rename:** Click **Rename** on a tab to change the tab name. You can cancel changes by clicking the `Cancel` button.
- **Sort:** Click **Sort:** on a tab, the Sort menu will appear. Drag the tab names up and down inside the Sort menu to change the sort order of the tabs. When done, click **Save tab order**.

Dashboard Widgets

By default, the Overview tab contains above mentioned widgets (function boxes). You can add these widgets on custom tabs:

- Expired Secrets
- Favorite Secrets
- Out-of-Sync Secrets
- Recent Secrets
- Reports
- Request Management

Widget Types

Table: Dashboard Widgets

Widget	Description
Expired Secrets	Displays expired secrets.
Favorite Secrets	Displays secrets marked as favorites. Also appears on the Secret Navigation Slideout.
Out-of-Sync Secrets	Displays secrets that are out-of-sync–the heartbeat or RPC have failed.

Widget	Description
Recent Secrets	Displays the secrets viewed most recently. Also appears on the Secret Navigation Slideout.
Report	Displays a report. Click the **Report Category** list to select a report from the drop-down menu. One report can be displayed per widget. Click the title of the report to navigate to the Report View page.
Request Management	Displays any requests pending for the logged in user.
+ Add Widget	Default widget that appears on an empty custom tab. When clicked, adds a widget that is not currently displayed to the Dashboard. This widget's function is duplicated automatically when you add a new Dashboard tab. You cannot remove this widget.

The Search and Browse widgets cannot be rearranged. They always remain in the top left region of the tab.

Managing Widgets

The following operations are available (by clicking the **More Options** icon on a widget) for managing widgets:

- **Delete:** Delete the widget from this custom tab.
- **Refresh:** Update the information in the widget. This is not available for all widgets.

Help Menu

The Help Menu is available via the question mark icon on the top right. It includes links to:

- About - direct links to Delinea Technical support, Blog, Feedback/Feature Request, Forums, Knowledge Base, and Mailing List.
- Secret Server REST API Guide - standard Swagger / Open API documents as well as interactive online documentation of the Secret Server REST API interfaces.
- User Guide - direct link to Secret Server documentation.

Running Dashboard Bulk Actions

You can perform bulk actions from the Dashboard on multiple secrets:

1. Navigate to the folder containing the secrets you wish to perform a bulk actions on.

   You can also run a bulk actions on the All Secrets page.

2. In the **Folder Scope** dropdown select to display the secrets in the current folder only or in the subfolders too.
3. Click to select the secrets you wish to include. To check them all, check the check box in the column header row. The **Bulk actions** button appears at the top. Click **Bulk actions** to select the action.

4. Available bulk actions include:
    - Activate
    - Assign jumpbox route
    - Assign secret policy
    - Assign to site
    - Change password remotely
    - Change security options
    - Change share permissions
    - Convert secret template
    - Deactivate
    - Erase secrets
    - Heartbeat
    - Move to folder
    - Request access
    - Set privileged account
    - Toggle autochange
    - Update associated secrets
    - Update password requirements

Bulk actions may differ by Secret Server version.

Configuration Search

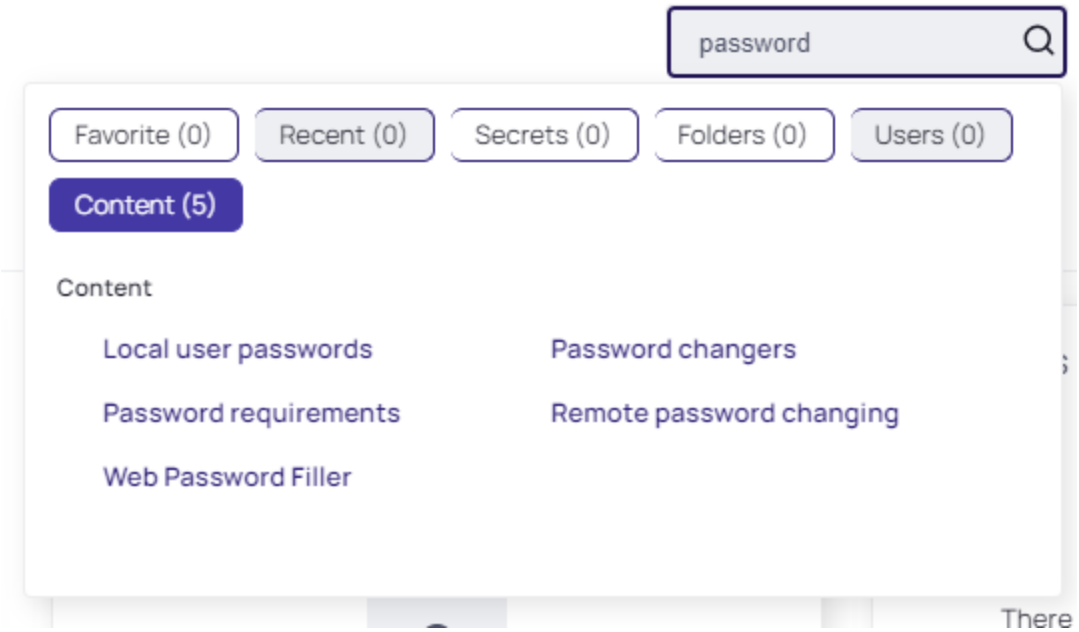
The Configuration Search (also called the "configuration preview") is a dynamic, one-stop locator for Secret Server configurations. To use it:

1. Navigate to **Admin> Configuration Search**. The Configuration Search page appears.
2. Either choose a configuration section from the links at the bottom or type a configuration name in the dynamic search box. Either way, the matching configuration sections appear at the bottom (replacing the original links).
3. Refer to the documentation topics for the specific configuration you navigated to.

Global Search

Global search appears at the top of every page in Secret Server. When you click inside the search box, a search window appears.

The window dynamically changes as you type a search term. At the top of the window there are six filter buttons that tell you how many search hits in each category. For example, if we typed "password," we might see this:



This tells us there are five places within the application (contents) that contain that word. The categories are:

- **Content:** The application interface.
- **Favorite:** Secrets marked as favorites.
- **Folders:** Secret folders.
- **Recent:** Recent search terms.
- **Secrets:** Secret names.

Main Navigation Drawers

Introduction

Since version 11.7, Secret Server's left navigation was greatly enhanced. The administration side panel was removed and replaced by a more all-encompassing drawer-like nested navigation panel.

You access it by clicking or hovering over the desired menu item or clicking the **>>** icon at the bottom. Clicking the **<<** icon closes the selected drawer.

Home Drawer

The home drawer contains the "Application Dashboard" on page 157.

Secrets Drawer

All Secrets

All Secrets displays all secrets that the user has permission to view. It provides a comprehensive list of all the secrets the user can access within Secret Server.

Quick Access

Quick access provides a convenient view of high-use secrets as well as keeping track of secrets that have been opened during the current session. It allows users to quickly find and access secrets they use frequently.

Favorites

Users can mark secrets as "Favorites" for easy retrieval. This feature allows users to quickly access secrets they use regularly without having to search for them each time.

Recent

This link shows the 15 secrets accessed most recently by the user. It helps users quickly return to secrets they have worked with in the near past.

Most Used

This link displays the secrets that the user has opened most often, providing quick access to the secrets that are most important or frequently needed in their daily tasks.

Shared with Me

This link shows the secrets that other users have shared with the user. It allows for collaboration and sharing of necessary credentials without compromising security, as permissions are still enforced.

Checked Out

The "Require Check Out" feature ensures that only one user at a time has access to a secret. When enabled, users must check out a secret before they can access it, preventing other users from accessing the secret while it is checked out.

Folders

The Folders section contains the "Folders" on page 1059

Discovery Drawer

Analysis

Opens the Discovery page to the Analysis tab. The analysis feature in Secret Server discovery provides a graphical summary of discovery activity, allowing users to manage discovered items effectively. It includes information about each item and is crucial for managing the discovered items. Items from disabled discovery sources will not be visible.

Network View

Opens the Discovery page to the Network View tab. The network view allows users to view the results of completed discovery after defining the discovery sources. It provides a graphical representation of the organizational structure and allows users to filter items by type, such as computers, computer accounts, public keys, service accounts, or directory accounts. Users can also initiate a rescan of certain items like computers by clicking the scan button.

Sources

Opens the Discovery page to the Sources tab. Discovery sources are the starting point for the discovery process. They define where and how Secret Server should look for accounts and dependencies.

Log

Opens the Discovery page to the Log tab. The discovery log provides a detailed record of the discovery process. It logs the activities and events that occur during the discovery scans, including the identification and categorization of computers, accounts, and dependencies within the network. The log is useful for troubleshooting, auditing, and understanding the actions taken by the discovery feature.

When a discovery scan is run, whether manually or automatically, the discovery log captures information such as:

- The start and end times of the discovery scan.
- Any errors or issues encountered during the scan.
- Details about the objects that were discovered, such as computers and accounts.
- Actions taken by the discovery process, such as creating new secrets for discovered accounts or linking discovered dependencies to existing secrets.

By reviewing the discovery log, administrators can gain insights into the effectiveness of the discovery scans, identify any potential issues that need to be addressed, and ensure that the discovery process is functioning as intended. It serves as a historical record that can be referenced to verify the discovery activities and outcomes.

Computer Scan Log

Opens the Discovery page to the Computer Scan tab. The computer scan log shows logs of the computer scanning process. This log provides details about the scanning of each machine found during the discovery scan to collect information configured to be collected by the discovery source.

Computer Scan Results

Opens the Discovery page to the Computer Scan Results tab. The computer scan results feature provides detailed information about the scanning process. It includes the results of the computer scan, which attempts to collect information such as local accounts, Windows services, scheduled tasks, and IIS application pools from each machine found during the discovery scan.

Reports Drawer

Report List

Opens the Reports page to the General tab, which contains the report list. The list provides a set of standard reports that include a variety of 2D and 3D charting and graphing components, as well as full grids of data. Some reports are purely data detailed and have no charts. Users can create custom reports based on Secret Server data such as user, audit, permissions, and folders. Reports can be organized into categories to aid in organization and access control.

Security Hardening

Opens the Reports page to the Security Hardening tab. The security hardening report checks aspects of Secret Server to ensure security best practices are being implemented. It identifies potential security issues within an installation and provides recommendations for hardening the security posture of the Secret Server environment.

User Audit

Opens the Reports page to the User Audit tab. The user audit report feature in Secret Server displays every password or secret accessed by a user within a specified period. This report is essential for assessing and controlling vulnerability risk when someone leaves the organization and for complying with regulatory requirements.

Schedules

Opens the Reports page to the Scheduled Reports tab that allows users to set reports to be generated on a regular schedule and sent via email to the management team or auditors. With the Health Check option, users receive an email only when a report has content, targeting unusual events for immediate alerts.

Audit

Opens the Reports page to the Audit tab. Secret Server maintains an immutable audit log of privileged user activity, which can be reviewed by the IT and security team. The audit feature includes customizable alerts (Event Subscriptions) that send email notifications when specified actions are performed or events occur. Audit reports provide accurate details on the secret itself, allowing users to monitor the level of activity on any secret they have access to.

Access Drawer

Users

Opens the Users tab of the User Management page. Secret Server supports both Local and Domain user accounts for authentication. Local user accounts are stored and managed by Secret Server, while domain user accounts are managed by Active Directory but subject to changes made in Secret Server. Administrators can create, edit, or remove local user accounts and configure account lockout settings to prevent repeated unsuccessful login attempts.

Groups

Opens the Groups tab of the User Management page. Secret Server allows administrators to manage users through user groups. Users can belong to different groups and receive the sharing permissions and roles attributed to those groups. This setup simplifies the management of permissions and roles that can be assigned to a user. Groups can also be synchronized with Active Directory.

Roles

Opens the Roles tab of the Roles Page. Role-Based Access Control (RBAC) in Secret Server enables IT admins to control what individual users can do within the application. Secret Server ships with out-of-the-box roles for common configurations and allows custom roles to be created that correspond to an organization's structure.

Directory services

Opens the Domains tab of the Directory Services page. Secret Server integrates with Active Directory, allowing administrators to automatically grant and revoke access to Secret Server with existing tools and policies. By assigning access based on security groups, manual permission granting is minimized, and users' rights in Secret Server change appropriately when their roles change.

IP address restrictions

Opens the Overview tab of the IP Address Management page. IP address restrictions in Secret Server allow administrators to control the locations and networks from which users can gain access. This feature enables limiting access to Secret Server to users who are "on network" and not accessing through VPN or other external networks.

Inbox Drawer

Secret Access Requests

Opens to Inbox > Secret Access Requests. The secret access requests feature in Secret Server allows a secret to require approval before access is granted. When a user requests access to a secret, an email is sent to the approval group(s), notifying them of the request. Members of the approval groups can approve or deny the request,

and access can be granted for a set time period. This feature establishes a workflow model where users must request access, and approvals can be managed through the Secret Server interface or via email if configured.

Secret Erase Requests

Opens to Inbox > Secret Erase Requests. Secret erase requests are used when a secret needs to be permanently removed from Secret Server. This is typically done for regulatory compliance and is not intended for database cleanup. When a secret erase request is made, it must be approved by designated approvers before the secret can be erased after a specified date and time. The process ensures that irreversible actions are taken with proper authorization and for valid reasons.

Application Requests

Opens to Inbox > Secret Application Requests. Application requests are not explicitly mentioned in the provided documents. However, in the context of Secret Server, this could refer to requests related to application account secrets or requests for integrating applications with Secret Server. These requests would likely follow a similar approval process to ensure proper access control and security compliance.

System Alerts

Opens to Inbox > System Alerts. System alerts are notifications that inform administrators and users about important system events or conditions that require attention. These can include alerts about system health, configuration changes, heartbeat failures, or other significant system-related issues. Alerts can be customized and sent via email to ensure prompt awareness and response.

Notifications

Opens to Inbox > Notifications. Notifications can include alerts about access requests, system events, and other important information that users need to be aware of.

Rules (Management)

Opens the Rules tab of the Inbox > Notifications page. Inbox rules are used to trigger actions based on notifications and can send either an email or a Slack message to a specified group of users when certain conditions are met. These rules are primarily for non-admin end-user communications and event subscriptions. They are used to filter notifications and define who receives the email or Slack message when a notification arrives.

Templates (Management)

Opens the Templates tab of the Inbox > Notifications page. Inbox templates are used for formatting and populating the content of messages sent to users' inboxes and can also be used by event pipelines. These templates define the subject, language used, and the canned text for the message, which contains variables that are replaced by Secret Server when the message is sent. Inbox templates can be system templates, which are read-only and can be cloned to create custom templates, or they can be custom templates that users create and edit themselves.

Inbox templates are crucial for managing how email and notifications are sent and received by users. They allow for configuration of notification scheduling, collecting notifications into digest format, and creating message templates and rules. This customizable tool set helps reduce alert fatigue by enabling users to receive notifications in a way that is most relevant and actionable for them.

Resources (Management)

Opens the Resources tab of the Inbox > Notifications page. Resources associated with inbox templates are items, such as images, that go along with any email based on the template. These resources are included in the message to enhance the visual presentation and provide additional context or branding.

Settings Drawer

All settings

Opens the Admin > All Settings page in the alphabetical view. This page serves as a comprehensive list of all system settings, organized alphabetically for easy navigation. This page allows administrators to quickly locate and manage various configurations, from authentication methods to email settings, ensuring that they can fine-tune the Secret Server environment to meet their organization's specific security and operational requirements.

Configuration Search

Opens the Admin > Configuration Search page. This page is a utility within Secret Server that enables administrators to swiftly find any system configuration by entering search terms. This feature streamlines the process of locating specific settings among the multitude of configurable options, saving time and simplifying system management tasks.

Distributed Engine

Sites and Engines

Opens the Sites and Engines tab of the Distributed Engine page. This tab provides an overview and management interface for the distributed engines and their associated sites within Secret Server. This tab allows administrators to configure and monitor the engines that handle tasks such as remote password changing and discovery scans, ensuring efficient distribution of workload across the network.

Site Connectors

Opens the Site Connectors tab of the Distributed Engine section. The tab provides access to the list of all the existing site connectors with the option to enable, disable, edit or add new connectors.

Configuration

Opens the Configuration tab of the Distributed Engine page. This tab is where administrators can set up and modify the settings for distributed engines in Secret Server. This tab includes options for engine registration, task assignment, and communication settings, which are crucial for the engines to operate correctly and securely within the distributed architecture.

Log

Opens the Log tab of the Distributed Engines section. The tab provides the log of all engines' events with the exact date and time, and detailed event messages. Select the related site from the Sites dropdown on the top to view the log for the specific site.

Audit

Opens the Audits tab of the Distributed Engine page. This tab offers a detailed log of all activities related to distributed engines, providing administrators with a transparent view of engine operations. This audit trail is essential for security and compliance, as it records actions such as engine registration, task execution, and any configuration changes.

Proxying

SSH

Opens the SSH Proxy tab of the Proxying page. This tab allows administrators to configure the settings for SSH proxying within Secret Server. This tab includes options to enable the proxy, set the proxy port, and manage other

related settings, which are vital for securely routing SSH sessions through Secret Server and protecting endpoint credentials.

RDP

Opens the RDP Proxy tab of the Proxying page. This tab is dedicated to configuring the RDP proxying feature in Secret Server. Administrators can enable the RDP proxy, define the proxy port, and adjust settings to ensure that RDP connections are securely routed through Secret Server, enhancing credential security and session management.

SSH Terminal

Opens the SSH Terminal tab of the Proxying page. This tab provides configuration options for the SSH terminal within Secret Server. This tab allows administrators to customize the terminal settings, such as enabling command menus and setting up session recording, to ensure a secure and controlled environment for SSH sessions.

Endpoint Configuration

Opens the Endpoints tab of the Proxying page. This tab lists all the server nodes, sites, and engines that can be configured as proxies for SSH and RDP sessions. Administrators can manage which nodes act as proxies and configure their public host and bind IP addresses, ensuring that sessions are securely routed through the designated endpoints.

Remote Password Changing

Configuration

Opens the Remote Password Changing page. This page is where administrators configure and manage the remote password changing feature in Secret Server. This page allows for the setup of password changers, scheduling of password changes, and monitoring of password change activities, which are critical for maintaining password security across the network.

Password Changers

Opens the Password Changers Configuration page. This page is a central location for setting up and managing the password changers used in Secret Server. Administrators can add, edit, and delete password changers, as well as configure their settings to ensure that passwords are changed according to organizational policies and compliance requirements.

Dependency Changers

Opens the Dependency Changers page. This page is designed for administrators to configure and manage the dependency changers in Secret Server. This page enables the setup of dependencies for secrets, ensuring that related credentials are updated in sync when a password change occurs, maintaining consistency and reducing the risk of service disruptions.

Heartbeat Log

Opens the Heartbeat Log tab of the RPC Logs page. This tab provides a log of all heartbeat activities within Secret Server. Administrators can review the log to monitor the health and accessibility of secrets, ensuring that credentials are valid and services are operational, which is essential for maintaining system integrity.

Password Change Log

Opens the Password Change Log tab of the RPC Logs page. This tab offers a detailed record of all password change activities in Secret Server. This log is crucial for tracking the success or failure of password changes, troubleshooting issues, and ensuring that password policies are enforced across the organization.

Diagnostics

System Log

Opens the Logs tab of the System log section. The tab provides the log of all the machines' events with the exact date and time, log levels, and detailed event messages. Select the related log level from the dropdown on the top to view the the specific log level events.

Server Nodes

Opens the Server nodes section that allows to edit nodes and filter them by workloads and clusters. See Secret Server Clustering for more info.

Diagnostics

Opens the Diagnostics section. Se Diagnostics for more details.

General

Secret Policy

Opens the Admin > Secret Policy page. This page is where administrators create and manage secret policies in Secret Server. These policies define sets of rules that can be applied to multiple secrets, such as remote password changing and security settings, to enforce consistent security requirements across the organization's secrets.

Secret Templates

Opens the Templates tab of the Admin > Secret Template page. This tab provides a list of all secret templates available in Secret Server. Administrators can create, edit, and manage templates, which dictate the structure and policies of secrets, ensuring that sensitive information is stored and handled in a standardized and secure manner.

Event Pipeline Policy

Opens the Policies tab of the Admin > Event Pipelines page. This page allows administrators to create and manage event pipeline policies in Secret Server. These policies define the workflows and actions triggered by specific events, enabling automated responses to activities within the system and enhancing operational efficiency.

Event Subscriptions

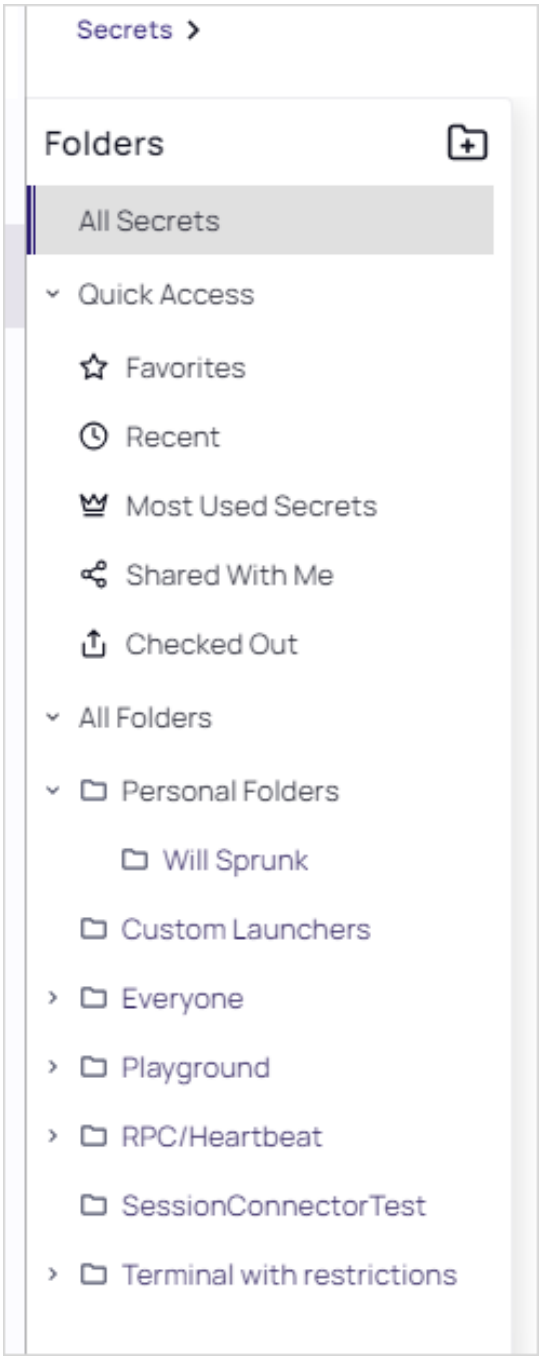
Opens the Subscriptions tab of the Admin > Event Subscriptions page. This tab is where administrators can set up and manage event subscriptions in Secret Server. This feature enables the configuration of notifications for defined events, which are sent to users' inboxes and can be further communicated externally via email or Slack, keeping stakeholders informed of critical activities.

Workflow

Opens the Admin > Workflows page. This page is dedicated to creating and managing workflows in Secret Server. Workflows allow for the setup of multi-tiered approval processes for secret access, providing a structured and secure method for sharing sensitive information within the organization while maintaining control and auditability.

Secret Folder-Tree Panel

The secrets menu item in the new UI has a collapsible secret folder-tree panel that occupies the full height of the page. This replaces the legacy folder tree at the bottom of the left navigation bar and delivers a significantly improved folder browsing experience. To view the panel, click the **>** icon on the bottom left of the page.



You can create folders, if you have permission to do so, by clicking the folder button on top of the folder panel.

Quick access provides a convenient view of high use secrets as well as keeping track of secrets that have been opened during the current session. Any pinned folder offers the same quick view but is scoped to the context of that folder. For example, only show me favorite or recent secrets from a single folder and its subfolders.

Direct links are available to common filtered views of secrets that allow for searching and advanced filtering within a specific context.

All folders to which you have access appear in the folder tree. A context menu offers options such as creating subfolders or pinning a specific folder. These options are also available when viewing a folder by clicking on the context menu next to the page title.

Secret Navigation Slideout

The Secret Navigation Slideout is a set of useful links to secret. Its appears on the top navigation bar.

Click the tab and the slideout appears:

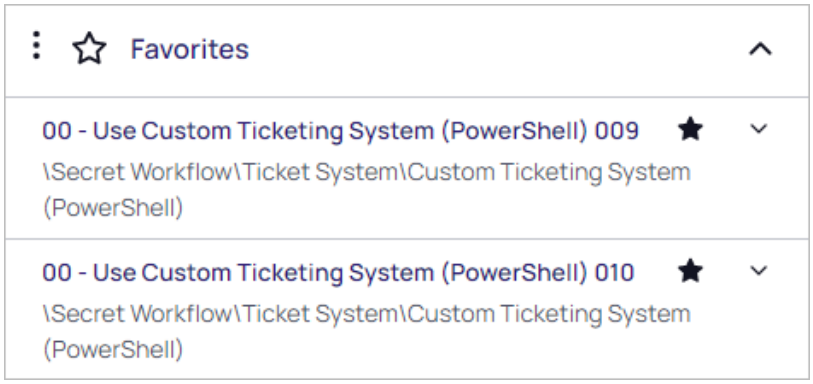
There are four dropdowns:

Favorites

These are the secrets you set as favorites by clicking the star icon on the secret's row on the All Secrets table.

Dropdown Quick List

When you click the dropdown, you see a list of your favorite secrets:



The initial view shows the folder path to the secret and a "favorite" toggle icon. You may need to hover over the blank space to see an unselected toggle.

> Some changes may require you to refresh the list by clicking the stacked dots icon.

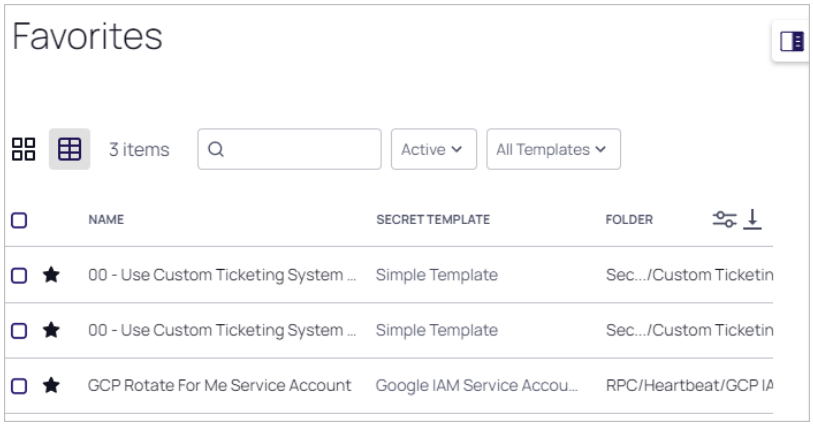
Each list item has its own dropdown. When you click it, a set of information about the secret appears, including a list of users or roles with access to (shared with) the secret.

> You may also have to enter a comment or check out the secret.

Most of the datums can be copied with a single click of the copy icon next to each. You can hover the mouse pointer over the "Shared with" list to see the type of access for that user or role, or you can click "See All" to see the entire list.

Favorites Page

Alternatively, in the slideout, you can click the Favorites title to navigate to the full Favorites page.



Recent

These are the secrets you have opened recently. The dropdown provided for each list item is exactly the same as that provided in "favorites," as is accessing the Recent page via clicking the title.

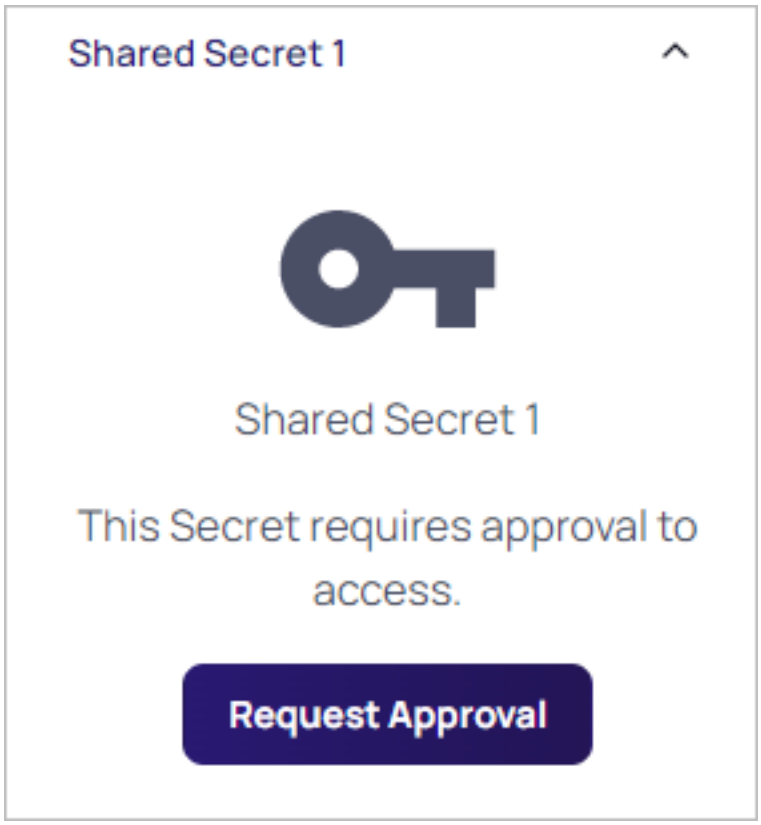
Most Used Secrets

These are the secrets you have opened the most often. The dropdown provided for each list item is nearly the same as that provided in "favorites," as is accessing the Recent page via clicking the title. The only difference is each list item states the time since last access.

Shared with Me

These are the secrets others have shared with you. No folder path is provided.

The dropdown provided for each list item provides a shortcut to request approval for the secret:



Administration

The Administration section covers a comprehensive set of administrative tools and configurations for managing Secret Server. It encompasses a wide range of functions, including database management (with options for changing SQL service account passwords and connections), security features (custom SSH cipher suites), service controls (enabling web services), data management (export and import settings), system integrations, maintenance capabilities, metadata handling, ticketing system integration, troubleshooting tools, and an unlimited administration mode.

Secret Server Architecture

Architecture diagrams are contained in the Architecture section of the Delinea documentation.

Application Settings

Go to **Settings > Configuration Search > Application** to get to the **Application Settings** page.

Alternatively, search for **Application** in the **Admin** search.

The Application Settings page allows configuring basic Secret Server settings. Here, you can enable automatic checks for software updates and gain early access to new features through the Early Adopter setting. You can also

opt to send anonymized system metrics to Delinea for product improvement and enable webservices to allow other applications to interact with Secret Server. Additional settings include configuring offline access duration for mobile devices, setting session timeouts for webservices, and managing token refreshes. Furthermore, you can enable Syslog/CEF Log Output, configure WinRM settings, and obfuscate personally identifiable information in audits.

Click **Edit** to enable, disable or set the desired configurations.

When done, click **Save** to save your settings.

Click **Test system log** to test your configurations.

The following configurations are available here:

- **Automatic checks for software updates**: Enable to be notified at the top of each page with a link to the latest update.
- **Early Adopter**: Enable to have access to the early adopter release - access, use, and benefit from the latest features, enhancements and improvements as soon as they are available. It is recommended to evaluate them in your test environment before an organization-wide rollout.
- **Send anonymized system metrics to Delinea**: Enable to sent anonymized data about your configuration and usage of Secret Server for the further product improvement. See Anonymized system metric information for more details.
- **Enable webservices**: Enable to allow other application to interact with Secret Server (requires to log in with the Secret Server credentials).
- **Maximum Time for Offline Access on Mobile Devices (days/hours)**: The maximum time for offline access on mobile devices setting in the server determines how long to cache secret data on the mobile device. Once the device is not in contact with the server for longer than the specified amount of time, the device removes its cache of the stored secrets. The only way to view secrets on the device once the cache is cleared is to connect to SS again so that the secrets can be re-downloaded. See Maximum Time Offline Explanation for more details.
- **Session timeout for webservices (days/hours/minutes)**: Set a Session time limit on Webservices API. Once Webservices expires, the user must log in again with their username and password.
- **Enable refresh tokens for webservices**: Tell OAuth2 to send back a refresh token during Authentication. This token will allow the user to get a new access token without having to enter credentials.
- **Maximum token refreshes allowed**: Set the maximum amount of times a user can refresh an access token.
- **Prevent direct API authentication**: Prevent non-Application Account users from directly authenticating against the API.
- **Prevent Application from Sleeping When Idle**: A keep alive thread will run in the background pinging the web URL to make sure IIS does not stop running due to inactivity.
- **Enable Syslog/CEF Log Output**: Check to enable Syslog/CEF Log Output. See Syslog/CEF Logging Advanced Settings Information for more details.
- **Syslog/CEF Server *** : Enter Syslog/CEF Server Address.
- **Syslog/CEF Port ***: Enter Syslog/CEF Server Port.
- **Syslog/CEF Protocol**: Select Syslog/CEF Protocol to use when sending logs.
- **Syslog/CEF Time Zone**: Select Time Zone to use when sending Syslog/CEF Protocol log entries.

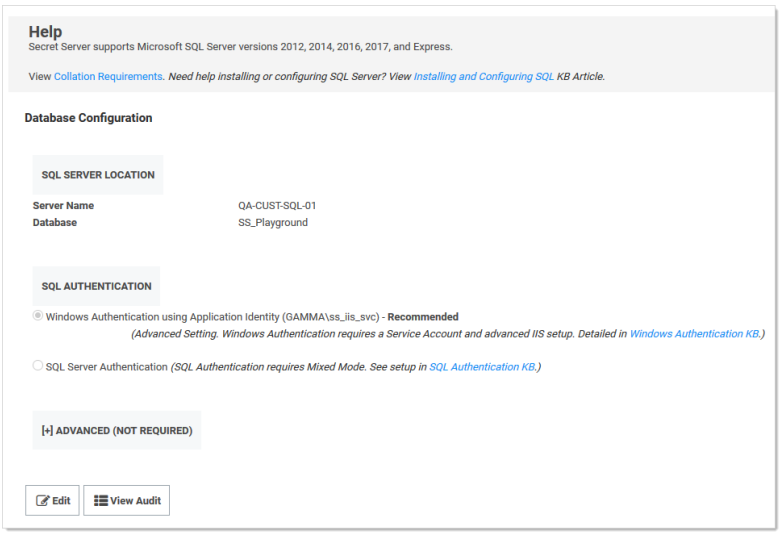
- **Syslog/CEF DateTime Format**: DateTime Format for Syslog/CEF Protocol log timestamps. Syslog: Jun 23 2022 11:22:33; ISO 8601: 2022-06-23T11:22:33.000.
- **Syslog/CEF Site**: This is the site that the CEF/Syslogs will run on.
- **Write Syslogs As Windows Events**: When enabled, Audits and Event Subscriptions will be written out to the Windows Event Log of the server.
- **WinRM Endpoint URL ***: The URL of the Windows Remote Management listener that will be used to run PowerShell scripts.
- **Enable CredSSP Authentication for WinRM**: Enable to allow a client to delegate credentials to a target server. See How do I configure CredSSP for WinRM? for more details.
- **Max Secret Log Length**: Enter the maximum Secret Log Length.
- **Custom URL**: The custom-set external binding for the server.
- **Privilege Manager Installation URL**: The custom-set location for Privilege Manager.
- **Obfuscate Personally Identifiable Information**: Delimit personally identifiable information (PII) in Audits. If obfuscate is selected, then it will automatically remove PII data from Audit exports. Note: This will only export data from the last enabled date.

Changing SQL Server Connection Parameters

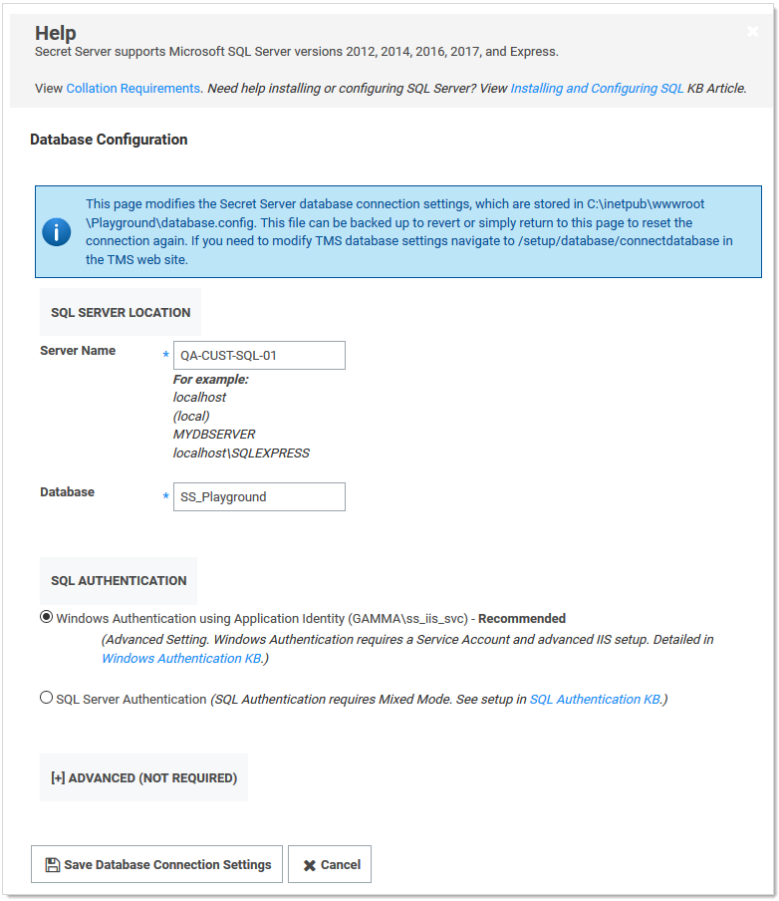
This topic only applies to **Secret Server On-Premises**.

Once Secret Server is installed, it may be necessary to change the connection string that Secret Server uses to connect to its database. You must be authenticated to access Secret Server and have the Administer Configuration role permission.

1. Click **Admin > See All**.
2. Type **Database** in the search text box and select **Database** in the dropdown list. The Database Configuration page appears:



3. Click the **Edit** button. The page enters edit mode:



4. Edit the parameters as desired.

5. Click the **Save Database Connection Settings** button. A confirmation message appears. Secret Server recycles its application pool (needed to clear the connection string cache), and then returns you to the Secret Server dashboard.

Change SQL Service Account Passwords without Restarting the SQL Service

This topic only applies to **Secret Server On-Premises**.

Introduction

Since Secret Server cannot restart SQL services, you can use the PowerShell script below to change SQL service account passwords without restarting the SQL service.

Requirements

This script uses a privileged account so the secret requires an associated secret added to it. No custom template is required for this changer.

Access will only be the Windows Server and not with SQL Server itself.

Procedure

Task 1: Adding the Script

1. Navigate to **Admin | Scripts**
2. Type or select the following:
   - Name: **SQL Server Service Password Rotation**
   - Description: **Script for rotating SQL Server service account- no restart**
   - Category: **Dependency**
   - Script: **Copy and Paste** the provided script below.
3. Click **OK**

Task 2: Creating the Dependency Changer

1. Navigate to **Admin | Remote Password Changing**
2. Navigate to **Configure Dependency Changers**
3. Click **Create New Dependency Changer**
4. Complete the form according to table below:

Field	Value
Type	PowerShell Script
Scan Template	Windows Service
Name	SQL Server Service Dependency Changer
Description	SQL Server service rotation - no restart
Port	Leave blank
Wait(s)	Leave at 0
Enabled	Leave checked
Create Template	Leave checked

5. Click the **Scripts** tab
6. Click the **Scripts** drop-down, and select the **PowerShell Script** dependency changer you just created.
7. Type `$MACHINE $SERVICENAME $PASSWORD`in the **Arguments** field.
8. Click **Save**.

Task 3: Adding the Dependency to a Secret

1. Navigate to the desired Secret.
2. Navigate to **Dependencies** tab
3. Click **New Dependency**.
4. Click the **Type** dropdown list and select the dependency created in the previous step (it should under **Standard** section).
5. Click the **Dependency Group** dropdown list to select a current group or create a new one. If creating a new one, provide the **New Group Name** and **New Group Site Name**.
6. Obtain the service name using following command on the SQL Server target machine:
   ```
   Get-Service *sql* | Select-Object Name
   ```
7. Type the service name in the **ServiceName** field.
8. Select a **Run As** secret if needed.
9. Type the for the target machine name in the **Machine Name** field.

You should now be able to rotate the password of the SQL Server service account and verify the dependency was successful.

Script: sqlservice-norestart-dependency.ps1

```
<# Utilize the Run As secret #>
$params = $args
$Target = $params[0]
$ServiceName = $params[1]
$ServicePwd = ConvertTo-SecureString -String $params[2] -AsPlainText -Force
$serviceCred = [pscredential]::new('Ignore this value',$ServicePwd)
Invoke-Command -ComputerName $Target -ScriptBlock {
    [pscredential]$cred = $using:serviceCred
    [System.Reflection.Assembly]::LoadWithPartialName
('Microsoft.SqlServer.SqlWmiManagement')> $null
    $sqlwmiLocal = New-Object 'Microsoft.SqlServer.Management.Smo.Wmi.ManagedComputer'
    $targetService = $sqlwmiLocal.Services | Where-Object Name -EQ $using:ServiceName
    if ($targetService) {
        try {
            $targetService.ChangePassword('ignore this value',$cred.GetNetworkCredential
().Password)
            $targetService.Alter()
        } catch {
            throw "Error updating the service password on $($using:Target) for service
$($targetService.Name): $($_.Exception)"
        }
    } else {
        throw "Service $($using:ServiceName) not found on $($using:Target)."
    }
}
```

Troubleshooting

SQLWMI is part of SMO with SQL Server.

> SMO stands for SQL Server Management Objects. It is a collection of objects designed for programmatic management of Microsoft SQL Server.

It exists on every installed SQL Server instance via the SQL Server Configuration Manager (SSCM). Newer versions of SQL Server no longer allow you to not install it. Just like WMI, there are cases where it can "break," most commonly when multiple versions of SQL Server are installed on the same machine–especially when a higher version was installed first, for example, if you installed SQL Server 2016 and then installed SQL Server 2012.

If running the dependency script you see the following error:

```
Failed to update <service name> on <target computer>. The following exception
occurred while trying to enumerate the collection: "An exception occurred in SMO
while trying to manage a service."
```

This can happen when SQLWMI is malfunctioning. Access that target and verify SSCM can be opened for that version of SQL Server you are trying to manage. If they have multiple versions, you need to open SSCM associated with the highest version found. If the target server is having issues, you will see WMI errors showing up when you open SSCM.

If this occurs you can solve the issue following this MS document: Error message when you open SSCM.

Custom SSH Cipher Suites

Overview

You can configure a custom cipher suite to assign to a site. The cipher is used for SSH client and server operations for that site, including SSH proxy, SSH terminal, discovery, remote password change, heartbeat and SSH scripts. Each cipher in the suite is prioritized by order used. Each one is tried before moving on to the next lower one if unsuccessful.

Please see "SSH Cipher Support" on page 875 for a listing of the default cipher suites.

Configuring Custom SSH Cipher Suites

Currently, you can only configure one custom SSH cipher suite.

To configure a custom SSH cipher suite:

1. Search for and access **Custom SSH Cipher Suite**. The Custom SSH Cipher Suite page appears.
2. The default **Details** tab provides a brief summary of the cipher suite. You can edit the name and description for the suite, as well as view the currently enabled algorithms.
3. Use the **Encryption Algorithms**, **Key Exchange Algorithms**, **MAC Algorithms**, and **Public Key Algorithms** tabs to enable, disable, and prioritize a list of each type of algorithm for the cipher suite.

   You can also check **FIPS** compliance for each of the algorithms. The types include encryption, key exchange, MAC, and public key algorithms.
4. To add algorithms to the list, click one of the algorithm tabs.
5. Select the **Edit** button on the top right of the algorithm page. The tab becomes editable.
6. Select the algorithms that you would like to add or uncheck to remove. When done, click **Save**.
7. To prioritize the algorithms, click **Reorder ciphers** on the algorithm page.
8. Drag the algorithms in the desired order. When done, click **Save Field Order**.
9. Select the **Audit** tab to display a list of actions taken with the cipher suite.

Using Custom SSH Cipher Suites

To enable the feature for a distributed engine:

1. Search for **Distributed Engine**. The Distributed Engine page appears.
2. Click the desired site name in the list. Its configuration page appears.
3. In the **Advanced site configuration** section click **Edit**.

If you cannot edit the page, you may need the "administer distributed engines" permission.

4. For the **SSH Cipher Suite** option, select **Use Custom Cipher Suite**.
5. When done, click **Save**.

Secret Server Database Maintenance

This topic only applies to **Secret Server On-Premises**.

Purpose

This topic, *Secret Server Database Maintenance*, is intended to share, at a high-level, information around best practices and recommended practices as it pertains to the Delinea product database hosted in Microsoft's SQL Server. The primary focus will be around SQL Server hosted on a physical or virtual machine.

However, a customer can choose *under specific architectures* to host the database within cloud-based SQL Server Platform-as-a-Service (PaaS) offerings such as Azure SQL DB or AWS RDS.

Customers and partners may use this document for reference with any version of SQL Server. Where there are differences in edition or limitations within PaaS services will be noted.

Implementation of database maintenance is not within the scope of Delinea Support.

Please note references or links provided to third-party or open-source tooling is not a direct endorsement by Delinea.

Intended Audience

SQL Server is a vast product with multiple components and configurations that Database Administrators can spend years to understand fully. This document is intended as a guide to get started; it is not a one-stop-shop for all the answers. Any job role can utilize this document as a reference for Delinea application databases.

Cloud Providers

The following providers will be considered and noted in this document:

- Azure: Azure SQL DB
- AWS: SQL Server on AWS RDS
- GCP: Cloud SQL for SQL Server

Azure SQL includes SQL Server services on Azure VMs and Azure SQL Managed Instance (MI). This document considers those services to be in parity with SQL Server running on-premises in the area of database maintenance.

Cloud provider information is subject to change by the specific provider and is out of the control of Delinea. This document will be maintained for significant changes, but advise consulting the provider's documentation to verify updates to those services.

Required Maintenance

SQL Server can be installed without much difficulty. Going through the wizard, you can do "next, next, next" and have SQL Server running on any server or desktop (even IoT devices). However, the wizard installation does not set up maintenance on the application databases you add. The application may work well initially; it starts with a small amount of data in the database. As your user base or usage grows, so will the database.

Ensuring proper maintenance is set up for the application database(s) is essential for the application and the business. As the enterprise expands the use of Delinea products, the hosted database expands and grows with data. Proper maintenance can ensure as the amount of data grows, the database is maintained correctly for the best performance possible.

The following sections will be the focus of this document. Links will be provided as needed to given additional resources to expand or research further with a given feature or concept of SQL Server.

Backups

If the application database is "lost" for whatever reason, how much data can you lose? This question needs to be answered by your backup plan and determines how you configure it within SQL Server. Backups in SQL Server serve the role of recoverability but also with database space management.

This section will go over the available backup types and how they offer recoverability. Additionally, some tools and options for automating those backups will be provided as examples.

Database Integrity

One of the most primary causes of data loss is hardware failures. Some hardware failures can cause SQL Server databases to become corrupt and unrecoverable without a backup being restored.

Detecting these scenarios will ensure your application and business continue operating or can recover safely.

An important note is that some database corruption can follow your database backups, so it is crucial to perform.

Indexes and Statistics

Users are starting their day, and the help desk begins to get calls that the application is not responding, or pages are taking ages to load. On average, the performance issues with application databases come down to two objects: indexes and statistics. The query optimizer in SQL Server is the piece that ensures the queries from the application run as efficiently as possible. A central part of that work is the indexes and statistics of the tables referenced in the query. Ensuring proper maintenance is performed regularly will help the query optimizer with its goal of efficient query processing.

Database File Size

The files of a database themselves do not require direct maintenance. This section focuses on the configuration of database files and how some maintenance tasks mentioned previously can help you manage the physical file size.

This will not apply to any PaaS offering of SQL Server; the cloud-provider maintains the files and underlying storage.

The Database

This document is focused on maintenance for Delinea database(s), but before we get to that, we need to get on the same page with what makes up a database in SQL Server. At a high overview, databases contain the following:

- Collection of **tables**
- The tables have **rows** that include various **columns**
- Those **columns** store certain types of data: dates, strings, binary, numbers, etc.
- The data is persisted to **database files** on a file system

While data is persisted to the database files, it will exist within memory when applications reference and manipulate it. SQL Server manages that memory and how/when that data gets persisted to those database files.

Additional objects you find in a database that are directly associated with the tables for performance are the following items:

- Indexes
- Statistics (index and column)

Physical Files

There are two types of database files with SQL Server:

- Data files (**.mdf* or **.ndf*)
- Log files (**.ldf*)

As mentioned previously, mechanisms within SQL Server handle persisting transactions that have written new data or modified current data into the data file. The log file stores a history of those transactions until the change has been persisted in the data file. If SQL Server crashes, that log file's job is to help get the database back to a consistent state.

Recovery Model

A central area to understand with database maintenance is the recovery model of the database. The recovery model will play a role in how SQL Server manages transactions being written between the log file (history of transactions) and the database's data file. Transactions being the meat of an application database, it plays a significant role. For example, the backups occur within the context of the recovery model. A database can only be set to one of three recovery models:

- Full
- Simple
- Bulk-logged

The default recovery model in SQL Server for new user databases is **Full**. Any new database that is created will be set to full recovery. There are possible model Database (system database in SQL Server) configurations that can change this. The below sections provide a bulleted list of points to be aware of with each recovery model type and how the log file space is managed.

Full

- Can recover to a specific point in time (minimal data loss)
- Log backups **are required**
- All transactions are fully logged
- Manage log space via backup and maintenance

Simple

- Recovery only to the end of the last backup (data loss greater)
- Log backups **are not allowed**
- Some transactions **will not** be fully logged
- Automatic log space management, handled by SQL Server
- Many HA features do not support use with this model (Log shipping, Availability Groups, etc.)

Bulk Logged

- Recovery to the end of the last backup (point-in-time recovery not supported)
- Log backups **are required**
- Bulk operations are minimally logged
- Manage log space via backup and maintenance

Additional details on SQL Server recovery modules can be found at Recovery models (SQL Server).

Cloud Providers

The database's physical files and the recovery model setting are areas of database management that the cloud provider maintains and controls. The databases are operating under the full recovery model for PaaS offerings of SQL Server.

Backups

Backups are always a best practice, even when you do not think you will need to restore from them. The unknown, after all, is unknown, and backups can always help plan around the unknown. Three types of backups can be taken in SQL Server.

Backup Types

Full

A full backup contains the full extent of data within the database and the transaction log to recover the database to a consistent state. This backup is required as a minimum before the other backup types can be done. During recovery, the full backup is the first backup type that has to be applied.

Differential

The differential backup contains all changes in the database *since* the last full backup. As databases grow, the time required for a full backup may extend, and it may not be reasonable to frequently take a full backup. A differential can be used to continue the frequency within a shorter time window. Additional differential backups will grow based on data changes since the last full that was taken.

Log

This backup contains changes logged in the database transaction log file that is not persisted in the database files. The size of these backups will vary based on the activity that has occurred in the database.

Recoverability

A backup strategy or schema should focus on the need for recoverability. A backup's primary purpose is to recover data. Delinea's database(s) recovery needs are purely based on the business requirements of a customer. A disaster recovery plan should account for and determine the backup scheme chosen based on those requirements.

Example

A retail company utilizes Secret Server and has labeled it a business-critical system. The product is only used by their users during regular operating hours of 7:00 AM to 5:00 PM Monday through Friday.

The business requirements for databases with systems labeled as business-critical have an RPO (recovery point objective) of 10 minutes. An example database configuration and backup scheme recommended in this situation that would meet the RPO would be as follows:

1. The database recovery model is set to full.
2. Backups are taken:
   - Full: weekly, every Monday @ 4:00 AM
   - Differential: twice daily, Monday - Friday @ 5:00 AM and 1:00 PM
   - Log: daily, Monday - Friday every 10 minutes between 5:00 AM - 6:00 PM

On Wednesday at 3:45 PM, the disk hosting the database server for Secret Server goes offline. When the disk and server are brought back online, the database is corrupt and cannot be brought online. The following backup files are needed to meet the RPO of recovering the data within 10 minutes of the time the database went offline at 3:45 PM:

- Full backup from Monday
- Differential backup from 1:00 PM on Wednesday
- All log backups from Wednesday at 1:00 PM to 3:40 PM

The above recovery meets the RPO as there is only a data loss potential of 5 minutes.

Backup Automation

Backup Types

Delinea products, such as Secret Server, offer the ability to schedule *full* backups daily. That option can be used if the recovery limitations around that schedule and backup type meet your business need. When you need to get more granular with the backup type and schedule, SQL Server does offer two options:

SQL Server Maintenance Plans

- These offer a "quick-and-dirty" method of getting backup jobs in place
- It can be difficult at times to troubleshoot issues with them.
- They do not offer much control; as databases are added or removed, you have to adjust the backup task manually.

SQL Agent Jobs

- These offer a more robust method
- Does require scripts be written to perform the backup
- Some tooling is available with prewritten scripts (see more below)

Ola Hallengren's Backup Script

This script is a standard tool used by many in the SQL Server community. You can download the script and find full details at SQL Server Backup. The stored procedure offered allows you extreme granular control over the backup type, what flags are used in the backup command, cleaning up old backup files, and stripping backups for performance improvements.

The scripts provided below can be used in a SQL Agent Job to be scheduled on the frequency required to meet your recovery needs.

Full Backup

```
EXECUTE dbo.DatabaseBackup @Databases = 'SecretServer',
@Directory = '\\backups\SQLServer\dbserver1', @BackupType = 'FULL',
@Verify = 'Y', @Compress = 'Y', @CheckSum = 'Y', @CleanupTime = 504,
@CleanupMode = 'AFTER_BACKUP'
```

The above command will take a full backup of the *SecretServer* database and write it to the network share `\\backups\SQLServer\dbserver1`. Also, backup verification and compression will be used, along with a checksum performed. If the backup is successful, full backups older than three weeks (504 hours) will be deleted from the network share.

Log Backup

```
EXECUTE dbo.DatabaseBackup @Databases = 'SecretServer',
@Directory = '\\backups\SQLServer\dbserver1', @BackupType = 'LOG',
@Verify = 'Y', @Compress = 'Y', @CheckSum = 'Y', @CleanupTime = 336,
@CleanupMode = 'AFTER_BACKUP'
```

The above command will take a log backup of the *SecretServer* database and write to the same network share. The same actions will be performed with backup verification, compression, and checksum. Log backups that are older than two weeks will be deleted.

Application Impact

Backups overall will not impact or impede the application from being able to perform write and read operations. You may observe spikes in resource usage on the database server under certain circumstances when the full backup is being run, but it should not be excessive.

It is advised to monitor performance during backup processing to ensure any performance issues or adjustments to schedules are addressed. As the database grows, it may require adjusting the backup scheme to meet the RPO without affecting the application's performance during business hours.

Cloud Providers

The cloud provider controls backups and the frequency of them. Specific controls around retention are specific to the cloud provider.

Azure SQL DB

- Azure SQL performs automatic backups (Automated backups in Azure SQL Database).
- Backup frequency is **full** every week, **differential** every 12-24 hours, and **log** backups every 5-10 minutes.
- Log backup frequency can also be based on computing size and amount of database activity.
- Backup storage is configurable, but the default configuration will replicate to another datacenter within that region.
- By default, backup retention can make a point-in-time recovery seven (7) days (configurable to a maximum of 35 days).
- Long-term backup retention is possible for up to 10 years.

AWS RDS - SQL Server

- Support for automated and manual backups
- "Backup window" is configured for the instance to include all databases, defaults to a 30- minute window
- Various situations cause the backups not to occur during the window. See AWS's Introduction to backups.
- Backup retention is configurable; the default depends on how the instance was created in RDS

GCP Cloud SQL

- Support for automated and manual backup
- The backup window is a 4-hour window, not configurable
- Recommended to perform manual backups on a more frequent schedule
- Reference documentation. See Google's About Cloud SQL backups.

Database Integrity

Various types of corruption can occur with databases, logical or physical. The corruption can be captured and travel with the backups of the databases as well. Corruption in the backup will cause issues when the need arises to implement a recovery plan, and there is no viable backup to use.

Identifying issues around database integrity early ensure the application's database stays online and backups are not affected.

Types of Corruption

Logical

This corruption involves consistency errors being found in the page-level structure of the database. Depending on where that page is found can determine how much effort is required to correct it or what amount of data recovery can be done.

Physical

This level of corruption is critical and will affect the uptime of the database. Physical corruption tends to be a hardware-level issue in most situations. In most cases, a backup has to be performed to recover.

Physical Corruption Warnings

There are two errors that SQL Server will write to the Windows Application event log that are excellent precursors to know physical corruption is possible.

Table: Corruption-Related Errors

Error 824	Error 825
This event occurs when data is read by the application and requires SQL Server to retrieve it from the physical file (data file). A consistency check is done when this process occurs, and if SQL Server detects an error, it will cause this event to generate. The event message will be similar to: *SQL Server detected a logical consistency based I/O error* You can find more details about this error at MSSQLSERVER_824 .	This error is more critical and should raise alarms (if being monitored). When the application queries data and SQL Server has to retry multiple times to retrieve it from the physical file, this event can be triggered. The event message will be similar to: *A read of the file '%ls' at offset %#016I64x succeeded after failing %d time(s) with error: %ls.* You can find more details about this error at MSSQLSERVER_825.

DBCC CHECKDB

CHECKDB is the command used for running integrity checks on any SQL Server database, including the system databases. You can review the syntax for the command at DBCC CHECKDB (Transact-SQL).

Execution Performance

Running an integrity check against a database is a process that can require additional resources compared to other maintenance tasks. Some tweaks can be done based on the version and edition of SQL Server to improve the performance. In Standard Edition, the CHECKDB command is a single- threaded operation, meaning as the database grows, the process will take longer to complete. In Enterprise Edition, the same command utilizes parallel

processing. In Enterprise Edition of SQL Server 2016, an option to configure the MAXDOP for the command is offered, allowing control of how many CPUs are used.

Additional details: CHECKDB From Every Angle: How long will CHECKDB take to run?

Application Impact

Internally when CHECKDB is executed against a database, it will utilize a snapshot of that database; this causes it to be an OLINE operation. This snapshot allows the operation to run without directly affecting on-going application activity against the database.

However, this task is an I/O intensive operation and should be done during minimal user activity in the application. While it won't directly affect the application, the overall high I/O activity may affect SQL Server's overall performance based on the resources the server is provided.

The CHECKDB command performs multiple operations, and those can be broken out (see documentation) and ran individually. An alternative that can allow less performant impact is to run those individual commands. A production database may only have CHECKDB run the physical checks only (*WITH PHYSICAL_ONLY*). It will allow the physical structure of the database to be verified for consistency. It is a low overhead check and will still check for common hardware issues that could cause data loss.

It is still recommended to perform a FULL run of CHECKDB on your production servers to ensure no type of corruption is found. Running this process on a restored database on a different server is fine, but it will not find corruption that may exist on that production server. If you restore a backup and run CHECKDB against that restored database, it merely verifies the backup you restored has no corruption in it.

Integrity Check Automation

Ola Hallengren's DatabaseIntegrityCheck Script (stored procedure)

This script is a standard tool used by many in the SQL Server community. You can download the script and find full details at SQL Server Backup. It will support the flags and options of CHECKDB by the version of SQL Server and is Availability Group aware.

The script below is commonly used for small databases, where integrity check is under an hour run time (total for all user databases).

```
EXECUTE \[dbo\].\[DatabaseIntegrityCheck\] @Databases = 'USER_DATABASES', @LogToTable = 'Y'
```

If the database is extensive in size, then a dedicated job for each database would be recommended. The script above can be updated for this purpose by replacing *USER_DATABASES* with the specific database's name.

Scheduling these jobs to run at least weekly is recommended.

Cloud Providers

Azure SQL DB

Azure utilizes various techniques to assist in managing data integrity. The Azure SQL engineering team provides more details at Data Integrity in Azure SQL Database. The article states running CHECKDB is unnecessary. The "what if" can still exist. Performing this maintenance is allowed on Azure SQL DB.

AWS RDS - SQL Server

The customer is responsible for running data integrity tests. The information above would be applied in the same way to the database hosted on AWS RDS SQL Server.

GCP Cloud SQL

The customer is responsible for running data integrity tests. The information above would be applied in the same way to the database hosted on GCP Cloud SQL for SQL Server.

Index and Statistics

The query optimizer lives to have good indexing and statistics in a database. An index strategy for a database will ensure the application performs optimally, at any size. Several chapters can be dedicated to indexing and statistics with SQL Server; it is a big topic for performance tuning. This section is going to cover a few areas to provide an overview.

Index Types

Index types vary on availability based on the version of SQL Server installed. The most common index types found in Delinea databases will be clustered and non-clustered indexes. If SQL Server 2016 SP1 is being utilized, then columnstore indexes may be found in the database.

The index and type found in a database can be gathered using the following query:

```
SELECT o.name, i.name AS indexName, i.type_desc, o.type_desc, ps.row_count FROM sys.indexes i
INNER JOIN sys.objects o ON i.object_id = o.object_id
INNER JOIN sys.dm_db_partition_stats AS ps ON i.object_id = ps.object_id AND i.index_id = ps.index_id
WHERE o.type_desc = 'USER_TABLE' ORDER BY o.name
```

More details on index types can be found at AI Skills Challenge: Indexes. Additional details on index architecture can be found in the SQL Server and Azure SQL index architecture and design guide.

Statistical Objects

Statistics are the lifeblood for the Query Optimizer in SQL Server. These objects are associated with columns and indexes. When an index is added to a table, a statistics object will be created. If there are any columns on the table used in aggregated reference or not included in an index, a column statistic object will be created. More details on statistics can be found at AI Skills Challenge: Statistics.

Fragmentation

Indexes will be the object type that deals with fragmentation. As you may use the alphabet or numbers to order a list, fragmentation in a table is when the physical order of that list no longer matches the logical order. Additional details on fragmentation can be found at Optimize index maintenance to improve query performance and reduce resource consumption.

Maintenance

The maintenance for indexes and statistics is intentionally split because it is recommended to perform them separately. From a pure performance perspective, when the database is relatively small (~under 50GB), both object types are not likely to take long; but it is purely based on SQL Server's resources. This type of maintenance is I/O intensive and can cascade to be memory and CPU intensive as the data footprint grows.

> **Important:** As the database grows, updating, adjusting, and monitoring this process are vital to getting the best performance out of SQL Server.

Indexes

Index maintenance covers fragmentation. Two types of maintenance tasks can be done with indexes: rebuild and reorganization. Each one has pros and cons:

Rebuild

Pros:

- Heavy resource usage for I/O on more extensive tables/indexes
- Potential to reclaim space in the data file and physical storage
- Updates index statistics object in the same transaction/processing

Cons:

- Resource intensive (I/O)
- The transaction cannot be killed without rollback occurring (that is, if the transaction runs for 2 hours, killing requires 2+ hours to rollback)

Reorg

Pros:

- Low resource usage
- The transaction can be killed without rollback
- It can pick up where it left off (to a degree), allowing for short burst during small maintenance windows
- Table locking is only on the current data page

Cons:

- Will not update statistics
- Based on the maintenance window, sizable indexes may take time to clear fragmentation

Statistics

The statistics objects exist for the columns and indexes on a table. These play a massive role with SQL Server's Query Optimizer. The statistics are how SQL Server knows how much data exist in a table. As the table grows, those statistics can become stale or out-of-date. SQL Server will update those statistics as they get marked out-of-date, but the method and schedule it does is not always optimal.

An example of how statistics work in SQL Server about data in a database can be found at Explaining SQL Server Statistics with Playing Cards.

Application Impact

When maintenance is performed on indexes and statistics, it is an I/O intensive process. For fragmentation to be cleaned up, the data has to be both read and written. If statistics have to be updated, the data has to be read; based on the sample rate, this can cause a higher workload based on data size.

When this maintenance is scheduled and implemented, it is recommended to monitor the process and periodically review the timing and workload it is causing on the SQL Server environment. There can be times when large tables need to be taken out of the regular schedule and have a dedicated schedule during a longer maintenance window or downtime for the application.

Index and Statistics Automation

Ola Hallengren's IndexOptimize (stored procedure)

This script is a standard tool used by many in the SQL Server community. You can download the script and find full details around this tool *here*. This procedure combines the ability to handle both indexes and statistics within one script, or you can create separate jobs when scheduling requires.

> The scripts provided below are simply samples and can be used for small-size databases. As the environment and data grow, the use of these scripts and configurations should be evaluated and tested.

The following T-SQL script can be used as a base template to manage both index reorg and statistics update in a single job (small environments):

```
EXECUTE \[dbo\].\[IndexOptimize\] @Databases = 'SecretServer', @FragmentationLow = 'INDEX_
REORGANIZE',
@FragmentationMedium = 'INDEX_REORGANIZE', @FragmentationHigh = 'INDEX_REORGANIZE',
@UpdateStatistics = 'ALL', @OnlyModifiedStatistics = 'Y',
@StatisticsSample = 80, @MSShippedObjects = 'N',
@TimeLimit = 28800, -- Time limit for work is 8 hours @LogToTable = 'Y'
```

Considerations on this command:

- This command's sampling rate is set to 80%; it is essential to monitor if this sampling rate is sufficient for the application's processing and workload.
- The TimeLimit set to 8 hours means when that limit is reached, no further commands on index reorg or statistic update will be issued; any current command executing will still have to complete.
- An additional parameter not specified, accepting the default value, is StatisticsModificationLevel. Please see the documentation link above for more details on this, if it needs to be used.

Cloud Providers

Azure SQL DB

Index and Statics maintenance is the responsibility of the customer.

AWS RDS - SQL Server

Index and Statics maintenance is the responsibility of the customer.

GCP Cloud SQL

Index and Statics maintenance is the responsibility of the customer.

Pruning Secret Server Log Data

> This section does not apply to cloud subscriptions.

Secret Server has processes that run as background processes, such as Discovery or Remote Password Changing, that frequently log the occurring activity. The majority of Secret Server features have a setting to control the data retention called *Days to Keep Operational Logs*. The default for this configuration is 30 days.

As well, the data retention feature was added to configure the Database Size Management. The management feature gives you a central location to control the data's max record age for particular Audit and Log/History tables. A number of the tables are included in this section as well.

The data is being stored in log tables within the database. These tables can commonly be the cause for the size of the database growing (logical space). Suppose your environment utilizes a large number of features in Secret Server heavily. In that case, you can benefit from having an external process such as a SQL Server Agent Job running to prune this data more frequently.

The following sections provide the query that can be used as a starting point for pruning each table as needed. It is recommended to adjust these based on each feature's usage and ***test*** before deploying to production instances.

> Table pruning varies whether it is based on a DATETIME or record count. Adjust the **@stopDate** or **@batchSize** as environmental needs require.

Initial Purge Processing

Before implementing the following example queries on a scheduled basis, you may find that an initial purge of log data is required in the tables mentioned below. To trim the tables to their initial desired size, large amounts of data may require deletion. Doing this in SQL Server can be done utilizing a batch method to query a certain amount, delete and then get the next amount until it does not find any more to delete.

To delete in batches is done using *TOP (<number>)* for the DELETE statement. The below can be used as a template to wrap around the DELETE statement of the above queries:

```
DECLARE @BatchSize INT = 4999 WHILE 1 = 1
BEGIN
DELETE TOP (@BatchSize)
/\* place log script here, after the DELETE \*/
-- Availability Group configuration in use (uncomment next line)
--WAITFOR TIME @waitTime
IF @@ROWCOUNT \< @BatchSize BREAK END
```

An example for the **tbSystemLog** the query would look like this:

```
USE \[SecretServer\] SET NOCOUNT ON;
DECLARE @MaxLogLength INT = 500 -- record count to keep DECLARE @BatchSize INT = 4999
WHILE 1 = 1 BEGIN
DELETE TOP (@BatchSize) FROM tbSystemLog
WHERE tbSystemLog.SystemLogId NOT IN ( SELECT TOP(@MaxLogLength) SystemLogId FROM tbSystemLog
ORDER BY tbSystemLog.LogDate DESC
,tbSystemLog.SystemLogId DESC
)
IF @@ROWCOUNT \< @BatchSize BREAK END
```

Example Pruning Code

tbStatusMessage Log Table

The bulk of logging for the activity occurring in Secret Server is being logged within this table.

Thread name examples: SecretItemHashUpdater, ActiveDirectoryMonitor, LegacyAgentMonitor, NodeClusteringMonitor, ConnectWiseMonitor, SSHProxyServer, ExpiredSecretMonitor, RPCHeartBeatMonitor, DiscoveryMonitor, ComputerScanMonitor, SecretComputerMatcherMonitor

Pruning code example:

```
USE \[SecretServer\]
SET NOCOUNT ON;
DECLARE @daysToKeep INT = 30 --days to keep of log
DECLARE @threadName varchar(250) = '' -- provide thread name to clean DECLARE @stopDate
DATETIME = DATEADD(DAY, -@daysToKeep, GETDATE()) DELETE
FROM tbStatusMessage WHERE
ThreadName = @threadName AND CreatedOn \< @stopDate
```

One consideration is utilizing the above as a template and control each *ThreadName* as your environment in different Agent jobs, that is, DiscoveryMonitor may contain more logs than RPCHeartBeatMonitor due to feature usage.

Do not remove the filter for *ThreadName* and do it purely on *CreatedOn*. There will be a varying amount of logs in Secret Server for each log that may affect the ability to troubleshoot.

tbComputerScanLastStatus Log Table

Pruning code example:

```
USE \[SecretServer\]; SET NOCOUNT ON;
DECLARE @daysToKeep INT = 30 --days to keep of log
DECLARE @stopDate DATETIME = DATEADD(DAY, -@daysToKeep, GETDATE()) DELETE
FROM tbComputerScanLastStatus WHERE LastScanDate \< @stopDate
AND NOT EXISTS (
SELECT NULL AS \[empty\]
FROM tbComputer AS c
WHERE c.ComputerId = tbComputerScanLastStatus.ComputerId
```

```
)
```

tbComputerScanLog Log Table

Pruning code example:

```
USE \[SecretServer\] SET NOCOUNT ON;
DECLARE @MaxLogLength INT = 1000 -- record count to keep
/\* IF EXIST - SQL Server 2016+ only \*/ DROP TABLE IF EXISTS \#TempComputerScanLog
;WITH CslRowNumbersByComputer AS (
SELECT ComputerScanLogId,ROW_NUMBER() OVER (PARTITION BY ComputerId ORDER BY ScanDate DESC
) AS RowNumber
FROM tbComputerScanLog
)
SELECT ComputerScanLogId INTO \#TempComputerScanLog FROM CslRowNumbersByComputer
WHERE RowNumber \<= @MaxLogLength
DELETE
FROM tbComputerScanLog
WHERE ComputerScanLogId NOT IN (SELECT 1 FROM \#TempComputerScanLog) GO
```

tbEventPipelineActivity Log Table

Pruning code example:

```
USE \[SecretServer\] SET NOCOUNT ON;
DECLARE @daysToKeep INT = 15 --days to keep of log
DELETE epa
FROM tbEventPipelineActivity AS epa
JOIN tbEventPipelinePolicyRun pr ON epa.EventPipelinePolicyRunId =
pr.EventPipelinePolicyRunId WHERE pr.QueuedDateTime \< DATEADD(DAY,-@daysToKeep,GETDATE())
```

tvEventPipelinePolicyRun Log Table

Pruning code example:

```
USE \[SecretServer\] SET NOCOUNT ON;
DECLARE @daysToKeep INT = 15 --days to keep of log DELETE
FROM tbEventPipelinePolicyRun
WHERE QueuedDateTime \< DATEADD(DAY, -@daysToKeep, GETDATE())
```

tbSecretViewTracker Log Table

Pruning code example:

```
USE \[SecretServer\] SET NOCOUNT ON;
DECLARE @daysToKeep INT = 1 --days to keep of log
DELETE
FROM tbSecretViewTracker
WHERE RecordedOn \< DATEADD(DAY,-@daysToKeep,GETDATE()) GO
```

tbOAuthExpiration Log Table

Pruning code example:

```
USE \[SecretServer\] SET NOCOUNT ON;
DECLARE @daysToKeep INT = 1 --days to keep of log
DELETE
FROM tbOAuthExpiration
WHERE ExpirationDate \<= DATEADD(DAY,-@daysToKeep,GETDATE())
```

tbSystemLog Log Table

This is the main log table for Secret Server. Normally, you would use this data for troubleshooting, but, if needed, you can prune it too. Pruning code example:

```
USE \[SecretServer\] SET NOCOUNT ON;
DECLARE @MaxLogLength INT = 500 -- record count to keep
DELETE
FROM tbSystemLog
WHERE tbSystemLog.SystemLogId NOT IN ( SELECT TOP(@MaxLogLength) SystemLogId FROM tbSystemLog
ORDER BY tbSystemLog.LogDate DESC
,tbSystemLog.SystemLogId DESC
)
```

tbSecretLog Log Table

Pruning code example:

```
USE \[SecretServer\]; SET NOCOUNT ON;
DECLARE @MaxDelete INT = 500
DECLARE @MaxLogLength INT = 1000 --record count on non-success
DROP TABLE IF EXISTS \#TempSecretNonSuccessLog
;WITH SecretLogRowNumbers AS (
SELECT TOP(@MaxLogLength) SecretLogId FROM tbSecretLog sl WITH (NOLOCK)
JOIN tbSecret s WITH (NOLOCK) ON s.SecretId = sl.SecretId AND s.Active = 1
WHERE sl.Status \<\> 'Success') SELECT SecretLogId
INTO \#TempSecretNonSuccessLog FROM SecretLogRowNumbers
DROP TABLE IF EXISTS \#TempSecretSuccessLog
;WITH SecretLogSuccessRows AS (
SELECT SecretLogId,
ROW_NUMBER() OVER (PARTITION BY sl.SecretId ORDER BY sl.DateRecorded DESC) AS rowNum FROM
tbSecretLog sl WITH (NOLOCK)
JOIN tbSecret s WITH (NOLOCK) ON s.SecretId = sl.SecretId AND s.Active = 1 WHERE Status =
'Success')
SELECT SecretLogId, rowNum INTO \#TempSecretSuccessLog FROM SecretLogSuccessRows WHERE rowNum
= 1
DELETE TOP(@MaxDelete) FROM tbSecretLog
```

```
WHERE SecretLogId NOT IN (
SELECT SecretLogId FROM \#TempSecretNonSuccessLog UNION
SELECT SecretLogId FROM \#TempSecretSuccessLog
)
```

Initial Purge Processing

Before implementing the above queries on a scheduled basis, you may find that an initial purge of log data is required in the tables mentioned above. To trim the tables to their initial desired size, large amounts of data may require deletion. Doing this in SQL Server can be done utilizing a batch method to query a certain amount, delete and then get the next amount until it does not find any more to delete.

To delete in batches is done using *TOP (<number>)* for the DELETE statement. The below can be used as a template to wrap around the DELETE statement of the above queries.

```
DECLARE @BatchSize INT = 4999 WHILE 1 = 1
BEGIN
DELETE TOP (@BatchSize)
/\* place log script here, after the DELETE \*/
-- Availability Group configuration in use (uncomment next line)
--WAITFOR TIME @waitTime
IF @@ROWCOUNT \< @BatchSize BREAK END
```

An example for the **tbSystemLog** the query would look like this:

```
USE \[SecretServer\] SET NOCOUNT ON;
DECLARE @MaxLogLength INT = 500 -- record count to keep DECLARE @BatchSize INT = 4999
WHILE 1 = 1 BEGIN
DELETE TOP (@BatchSize) FROM tbSystemLog
WHERE tbSystemLog.SystemLogId NOT IN ( SELECT TOP(@MaxLogLength) SystemLogId FROM tbSystemLog
ORDER BY tbSystemLog.LogDate DESC
,tbSystemLog.SystemLogId DESC
)
IF @@ROWCOUNT \< @BatchSize BREAK END
```

Availability Group Considerations

This section discusses specific considerations when performing database maintenance with Availability Group (AG) databases in SQL Server.

Backups

AGs allow backups to be offloaded to secondary replicas based on how the Backup Preference is configured. See Availability Group Properties: New Availability Group (Backup Preferences Page) for details. Full and Differential backups will be the only ones that can be performed on the secondaries (*only ones that support copy-only*). The Log backups can only be taken from the primary replica. As the database grows, this can be a consideration to offload that activity.

When utilizing the *DatabaseBackup* script and frequent log backups are being done (e.g., every 5 minutes), the cleanup of those log backups may need to be done in a different job. When a larger number of backup files need to be cleaned up, based on the *CleanupTime* configured, you may observe excessive waiting periods. If this begins to affect log backups not being done every 5 minutes, it is recommended to move that cleanup to a different job.

Transaction Log

The log file in a database joined to an AG is managed by SQL Server differently than a standard database. In an AG configuration where synchronous replicas are in use, the log transaction is not marked inactive until that transaction has been written on all log files for the database replicas.

Large transactions can affect this timing/latency. Maintenance such as Index rebuilds can cause large amounts of small transactions to occur. In that processing on an AG, the transaction won't complete until it is replayed on the log for all secondary replicas, in synchronous mode.

A common practice is to adjust those secondary replicas' the availability mode to asynchronous commit during maintenance windows. See Change availability mode of a replica within an Always On availability group for details. It can ensure the activity can run more efficiently on the primary replica and complete within limited maintenance windows. Once the redo log catches up on the secondaries, they can be set back to synchronous commit mode.

Shrinking Log Files

Scheduling or frequently shrinking the database files for a SQL Server database is not recommended. In situations where the above log management scripts were used for the initial pruning of a large amount of data, you can shrink the data file to reclaim storage space. There may be other occurrences that a shrink is warranted. If proper maintenance is in place for the regular database, shrinking database files can cause more performance issues as time goes on.

When you need to shrink **the data files**, be aware that it requires exclusive locks on the database tables and indexes. More care is needed when shrinking the data file to ensure no connections are present to the database to complete the operation without interruption. Doing this while monitoring the progress is recommended, especially if this has to be done during business hours or when the application cannot be brought down. The data file shrink is also an IO and CPU-intensive operation.

When you need to shrink **the log file**, it is less intrusive and will not cause locking, and generally is not a CPU-intensive operation. The log file shrink will only be able to reclaim the inactive portion of the log file. Steps such as taking a full backup and then a log backup, at times, can help get those active portions switched to inactive so you can shrink the log file.

Shrinking Availability Group Log Files

Database log file growth can happen on the primary replica and even secondaries. When it becomes excessive, it can be a cause for concern. If the log file on the primary is growing and taking action to shrink them are not working, check your secondaries' performance.

Secondary replicas in asynchronous-commit mode still need that transaction from the primary replayed to keep the data synced. When those secondary replicas get behind, it prevents the log from clearing on the primary, and continuous log growth will be observed. Troubleshooting will be required on those secondaries to determine why they are not keeping.

One area to check if you see continuous log growth on the primary is ensuring all the secondaries have data movement enabled. If SQL Server paused data movement, you could see log file growth on the primary. Once you

turn data movement back on, it will only be a waiting game until those secondaries are caught up. Once that happens, perform a full backup on the primary, and then the log file can be shrunk.

You can view the logical structure of the log file using *DBCC LOGINFO*. This command outputs a record for each virtual log file that SQL Server has created. The status column of that output will show you 2 (active) or 0 (inactive). The active records could mean several things: backup has not happened yet, active transaction, or waiting on replication.

Enabling Webservices

This topic only applies to **Secret Server On-Premises**.

Webservices can be enabled at the **Administration > Configuration** general tab. Enabling webservices simply makes the ASP.NET webservices built into Secret Server available. They are found under `/webservices/sswebservice.asmx` in your Secret Server directory. They run on the same port as the Web application. You can view them with a browser to see the functionality that is offered. Specific webservice functionality is documented in the Secret Server Webservice API guide.

Exporting and Importing Secret Server Settings

Overview

Secret Server can now export and import Secret Server settings as a JavaScript Object Notation (JSON) file. With this, you can more easily move settings from an existing Secret Server environment to another.

Prerequisites

Required General Permissions

There are the permissions required to access and perform the process. These are:

To view the Export/Import page or menu link:

- Administer Export or View Export
- Administer Import

To view audits (at least one is required):

- Administer Users
- Own User
- View Users

Exporting to a JSON file:

- Administer Configuration
- Administer Export

Importing from a JSON file:

- Administer Configuration
- Advanced Import

Required Additional Permissions

Some of the settings require additional permissions to export or import:

Table: Required Additional Permissions

Setting	Permission
OpenID Log on	Administer Delinea One
SAML	Administer Configuration SAML
Security	Administer Configuration Security
Session Recording	Administer Configuration Session Recording
SSH Commands	Administer SSH Menus
Delinea One Log on	Administer Delinea One
Two Factor Log on	Administer Configuration Two Factor

Required Licenses

Additional licenses may be required to import or export some settings.

Advanced Auditing License

This license is required for these settings in the Application Settings category:

- SyslogCefLogSite
- SyslogCefPort
- SyslogCefProtocol
- SyslogCefServer
- SyslogCefTimeZone

Enterprise Edition

> These settings are also available with the Professional Edition and Approval Workflow Add-on licenses.

The Enterprise Edition license is required for these settings in the Launcher Settings category:

- CheckInSecretOnLastLauncherClose
- CloseLauncherOnCheckInSecret

It is required for these settings in the Permission Options category:

- EnableApprovalFromEmail
- ForceSecretApproval

It is required for the TicketSystems category.

Pro Edition

The Pro Edition license is required for these setting categories:

- SAML
- Session Recording

Platinum Edition

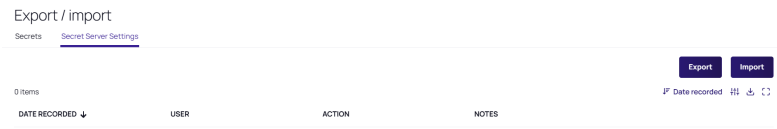
The Platinum Edition license (or Pro Edition and Unix SUPM licenses) is required for the SSH Commands setting category.

Procedures

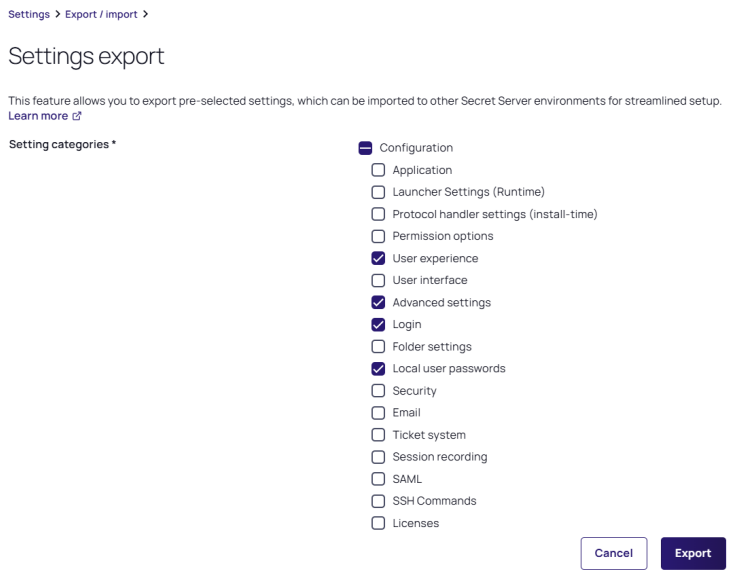
Exporting Settings

To export Secret Server settings:

1. Go to **Admin > Export/Import**, and select the Secret Server Settings tab.



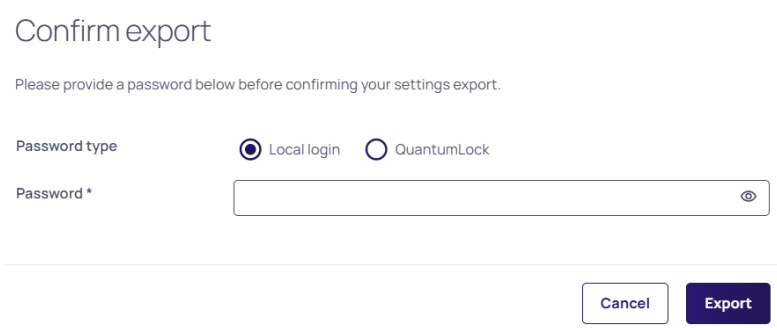
2. Click the **Export** button at the top right. The Settings Export page appears:



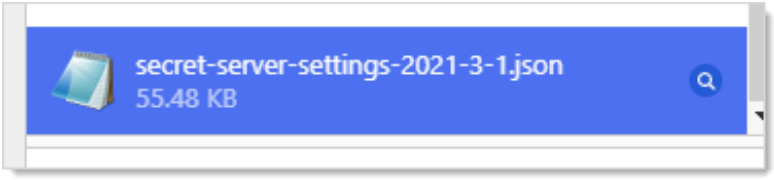
3. Click to select the check boxes for the settings categories you wish to include. Clicking the **Configuration** check box selects or unselects all available categories.

> See the Setting Category Reference section for details on the settings in each category.

4. Click the **Export** Settings button. A Confirm Export popup appears:



5. Click to choose **Local Login** or **QuantumLock**.
6. Type your password in the **Password** text box.
7. Click the **Export** button. The JSON file appears in your browser's downloads:

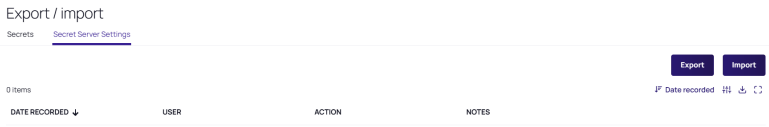


> This example is for the Vivaldi Chrome browser—yours will likely look different.

Importing Settings

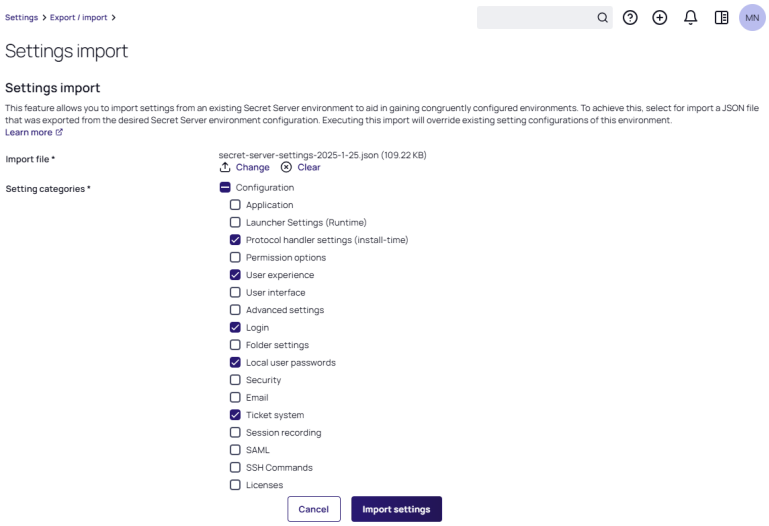
To Import Secret Server settings:

1. Go to **Admin > Export/Import** and select the Secret Server Settings tab.



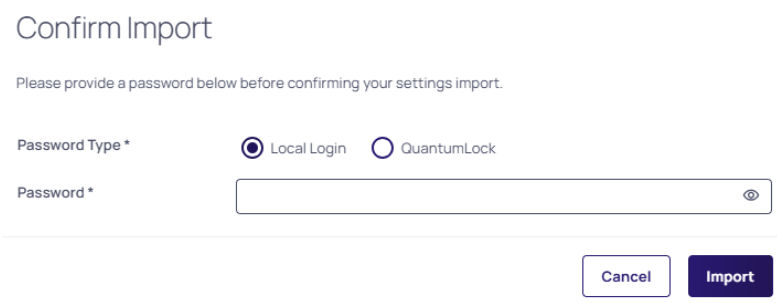
2. Click the **Import** button. The Settings Import page appears. Click the **Change** link, and navigate to and select the JSON file you want to import. The name of the file you chose appears above the Change link.

   Click to select the check boxes for the settings categories you wish to include. Clicking the **Configuration** check box selects all available categories.

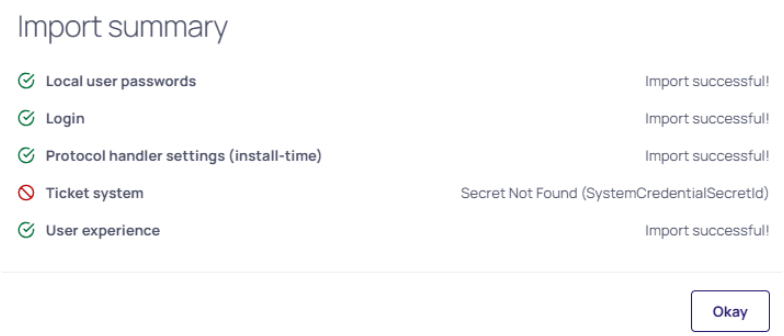


3.
> Some settings may not allow you to select them, based on your permissions and licenses. Another possibility is the category was not included in the original export. Hover the mouse pointer over any of these settings to view a hint of what is likely causing it. See the Setting Category Reference section for details on the settings in each category.

4. Click the **Import Settings** button. A Confirm Import popup appears:



5. Click to choose **Local Login** or **QuantumLock**.
6. Type your password in the **Password** text box.
7. Click the **Import** button. An Import Summary popup appears.



8. Click the **Okay** button. The importation appears in the log that you saw earlier on the Export / Import page.

Setting Category Reference

This section details what setting are contained in the following settings categories:

> Some settings are unavailable in certain environments or if requiring a license or permission.

Application Settings

These settings correspond to the Application Settings section on the Configuration General page.

This setting is unavailable in an on-premise environment:

- DisplayDowntimeMessageToAdminsOnly

These settings are unavailable in a cloud environment:

- AllowSoftwareUpdateChecks
- CustomURL
- EnableKeepAliveThread

- TmsRootUrl
- WriteSyslogToEventLog

This setting is unavailable in an IBM environment:

- AllowSendTelemetry

Advanced Settings

These settings correspond to the Advanced Settings section on the Configuration Advanced page.

Launcher Settings (Runtime)

These settings correspond to the Launcher Settings (Runtime) section on the Configuration General page.

Launcher Deployment Type setting can be one of the following:

- 0: Click Once
- 1: Protocol Handler

This setting is unavailable in a cloud environment:

- LauncherDeploymentType

Email

These settings correspond to the Email tab on the Configuration Email page.

These settings are unavailable in a cloud environment:

- SmtpDomain
- SmtpPassword
- SmtpPort
- SmtpServer
- SmtpUseCredentials
- SmtpUseImplicitSSL
- SmtpUserName
- SmtpUseSSL

Folder Settings

These settings correspond to the Folders tab on the Configuration Folders page.

Licenses

These settings correspond to the licenses listed on the Licenses page.

Local User Passwords

These settings correspond to the Local User Passwords tab on the Configuration Local User Passwords page.

Login

These settings correspond to the Login tab on the Configuration Login page.

These settings unavailable in a cloud environment:

- CacheAdCredentials
- TwoFactor.Radius.ClientPortRange

Permission Options

These settings correspond to the Permission Options section on the Configuration General page.

The Default Secret Permissions setting can be one of the following:

- 0: Secrets inherit permissions from folder
- 1: New Secrets copy permissions from folder
- 2: Only creator has permissions to new Secrets

Protocol Handler Settings (Install-Time)

These settings correspond to the Launcher Settings (Runtime) section on the Configuration General page.

SAML

These settings correspond to the SAML tab on the Configuration SAML page. To insert a new identity provider, in the same instance the export file came from, the IdentityProviderId setting must be set to 0. Otherwise, it will treat it as an update (see External Instance Id). The identity provider name cannot match another already in the database.

Security

These settings correspond to the Security tab on the Configuration Security page.

These settings are unavailable in a cloud environment:

- DatabaseIntegrityMonitoringSymmetricKey
- EnableDatabaseIntegrityMonitoring
- EnableHSTS
- FipsEnabled
- ForceHttps
- HSTSMaxAge

Session Recording

These settings correspond to the Session Recording tab on the Configuration Session Record page. The launcher must be enabled, and a valid license for Session Monitoring is required to export and import this feature.

These settings are unavailable in a cloud environment:

- ArchiveLocationBySite
- ArchivePath
- DaysUntilArchive
- EnableArchive
- EnableHardwareAcceleration
- StoreInDatabase
- VideoCodecId

To update SSHProxyRecordVideo or SSHProxyRecordKeyStrokes, SSH Proxy must be enabled.

To update RDPProxyRecordVideo or RDPProxyRecordKeyStrokes, RDP Proxy must be enabled.

SSH Commands

These settings correspond to the SSH command restrictions, the SSH commands, allowed command menus, and blocked command lists.

Ticket System

These settings correspond to the Ticket System tab on the Configuration Ticket System page. To insert a new ticket system in the same instance the export file came from, TicketSystemId must be set to 0. Otherwise, it will treat it as an update (see External Instance Id). The ticket system name cannot match another already in the database.

User Experience

These settings correspond to the User Experience section on the Configuration General page.

- Application Language
- Default Date Formats can be found in the tbDateOptions table.
- Default Time Formats can be found in the tbTimeOptions table.
- Default New User Roles can be found in the tbRoles table
- Server Time Zones can be found in the server registry: `Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones`

User Interface

These settings correspond to the User Interface section on the Configuration General page.

These settings are unavailable in an IBM environment:

- AllowUserToSelectTheme
- CustomLogoCollapsed
- CustomLogoFullSize

JSON Export File

In addition to the setting categories, here are a few components of the JSON export file that you should be aware of.

External Instance ID

```
"externalInstanceId": "95931fb9-02b0-47a5-a59d-69d6543a192d",
```

The external instance ID is an identifier for the Secret Server instance the settings were exported from. If you change this ID, Secret Server will assume the export came from another database and will insert new records for the ticket system (TicketSystemId) or SAML (Identity Providers–IdentityProviderId) categories. To add a new record in the same instance, set the ID to 0 and it will be treated as a new item.

Configuration Version

```
"configurationVersion": "1.0.0",
```

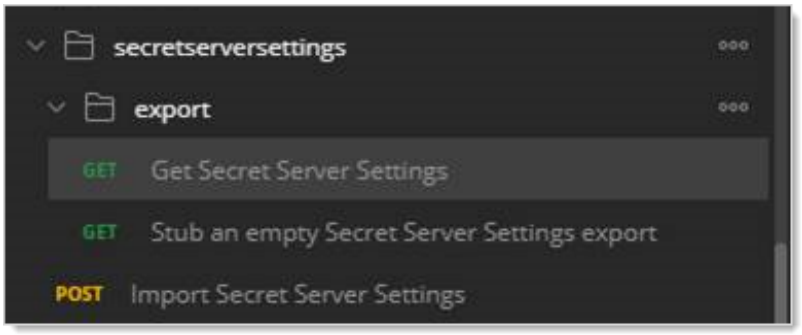
This is the configuration version the settings were exported from. In there future when other settings are added, it will help Secret Server determine which settings are available and which are not in the database.

JSON Import File

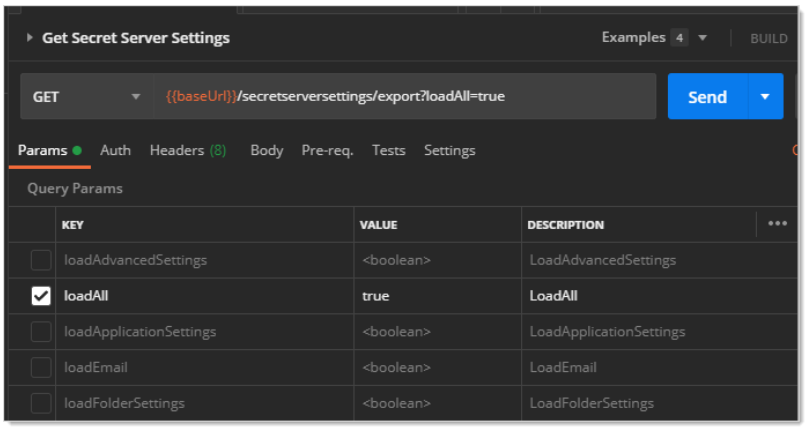
In the UI, the exported JSON file can be easily modified and used as the import JSON file. For the API, the exported JSON must be added to the data object. Then manually update the desired filter category load to true to import.

API Calls Filter

Secret Server has settings import/export endpoints for the API to manipulate. Opening Postman and going to **secretserversettings > export > GET Secret Server Settings**, you would see:



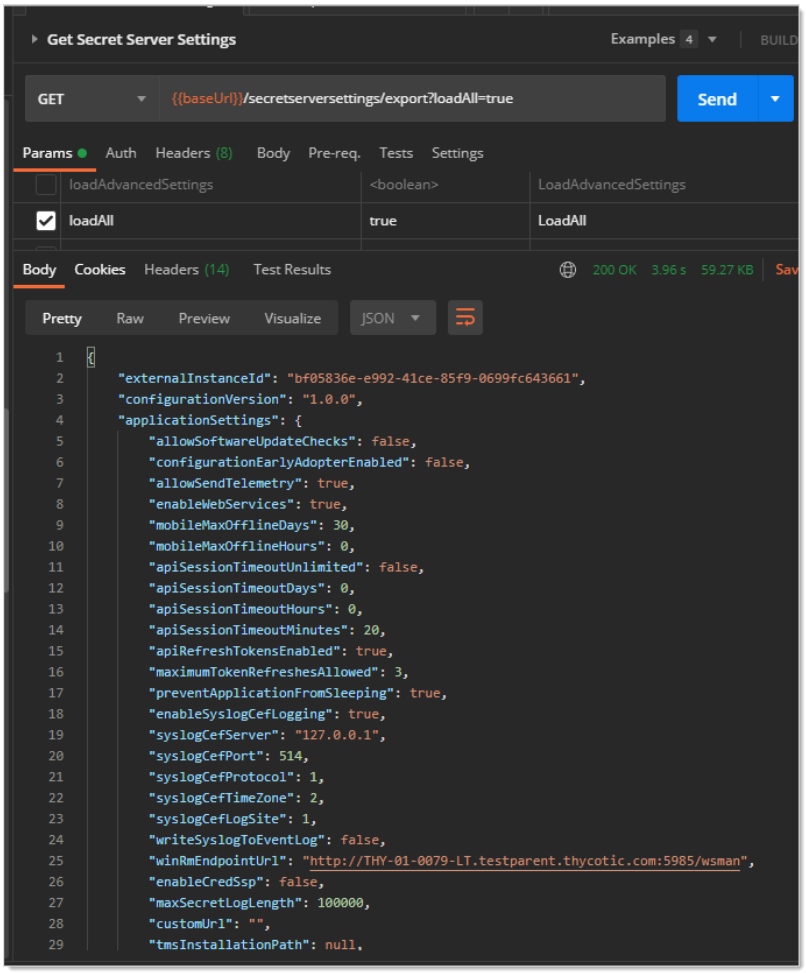
Looking at the query parameters for that endpoint, we see:



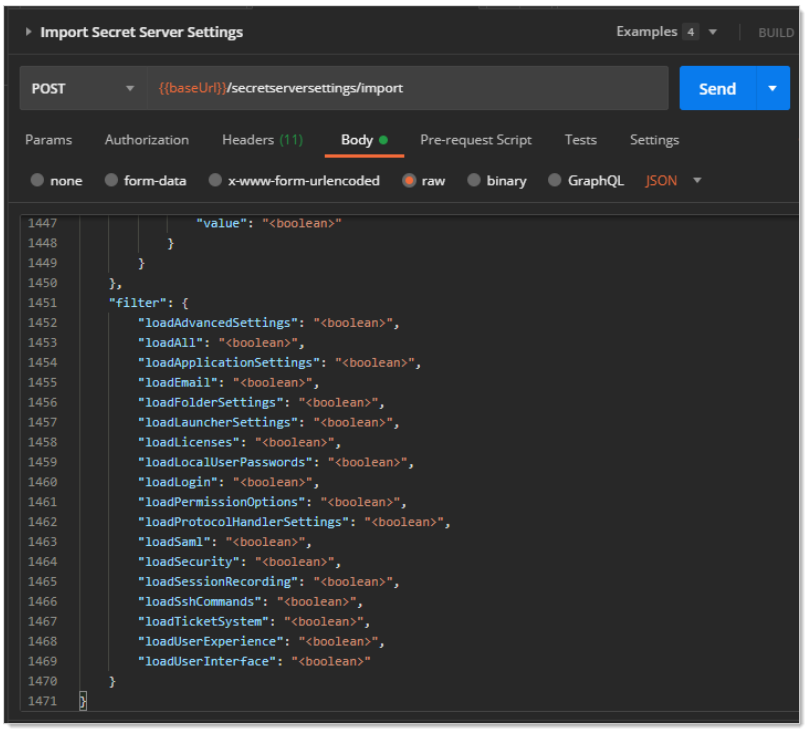
The keys are equivalent to those on the user interface or those in the JSON file.

The `loadAll` key tells Secret Server to update all the available settings. These include application settings, launcher settings, protocol handler settings, permission options, user experience settings, and user interface settings.

If you click the **Body** tab below, you can see what JSON code represents the key you chose for the export:



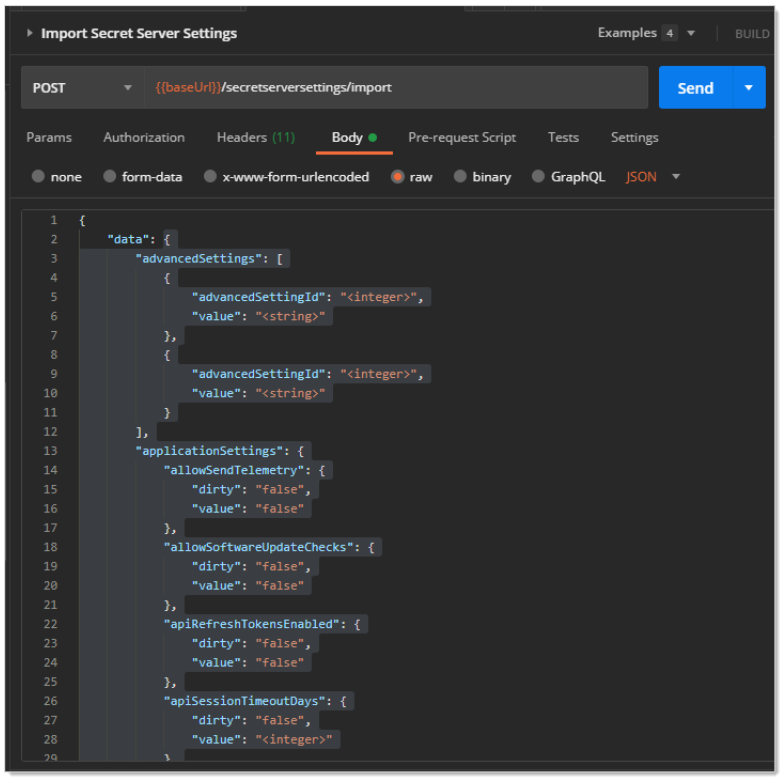
When using the POST Import Secret Server Settings command, you will see a filter object at the bottom of the code stipulating what to update:



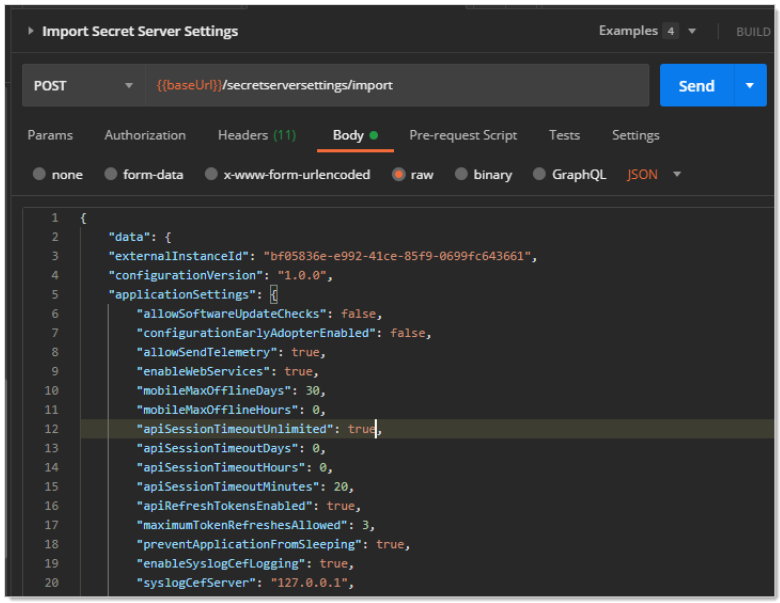
For example, if you set `loadApplicationSettings` to `true`, only the application settings are updated, assuming the objects stipulated were sent with the request. Similarly, included objects that are disallowed by the filter are ignored.

To make a GET call to update a single setting:

1. Import the category it belongs to. For example, if you want to update `apiSessionTimeoutUnlimited` to `true`, you would copy the entire `applicationSettings` result (the category and all of its settings).
2. For the POST Import Secret Server Settings call, remove the settings in the data section, leaving the filter section as is:



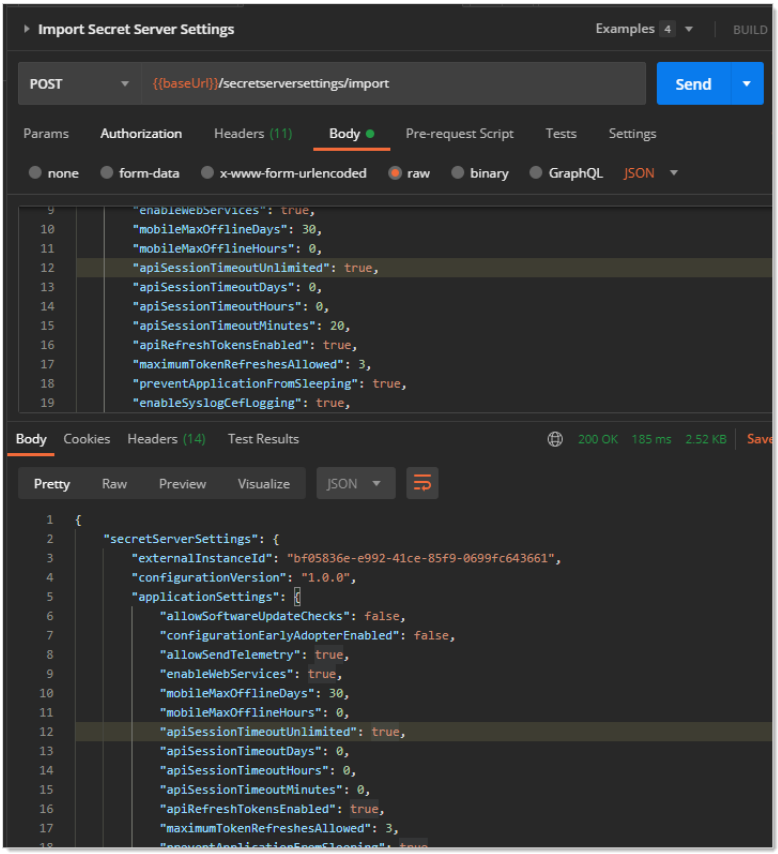
3. Paste the settings you copied earlier in its place.
4. Change the `apiSessionTimeoutUnlimited` setting to `true`:



5. Scroll down to the filter section and remove the filters you do not want to update. Alternatively, you can replace all the <bobolink> settings with `false` for the filters you do not want.
6. If you want to set a nullable field back to null, set the dirty flag and the value to null. For example:

```
"siteId": {
  "dirty": "true",  "value": null}
```

7. Click the **Send** button. If all goes well, Postman will return the updated category object:



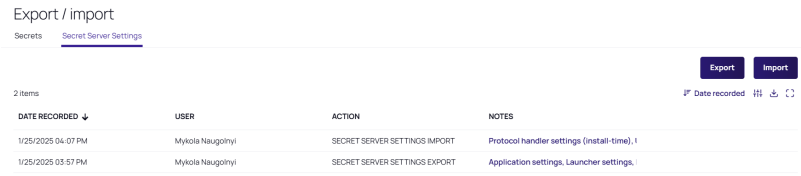
> **Note:** In this example, we copied the whole data object, but you do not have to. For a quick update, you can Import with just the settings you want to update. Anything not sent is ignored. This is the reason a nullable setting has to be explicitly set to null, along with setting the dirty flag–everything set to null is ignored, the same as if you did not send the setting at all.

8. If something went wrong, you will see an error section at the bottom of the results:

Audits

An audit is recorded for each setting category that was exported or imported by user. The individual setting audits can be viewed on the Configuration Audit page.

Figure: Audits on the Export / Import page:



If there are errors, a system log entry will also be saved with details:

- Email
- Login
- Security
- Ticket System
- SAML - Completed with errors

Events

When Secret Server settings are exported or imported, an SECRETSERVERSETTINGS event is logged.

Logs

When Secret Server settings are exported or imported or validation errors occur, a new log entry will appear in the `SS.log` file.

> `<Username>` and `<USERID>` are replaced with your values. The items in the parentheses are the errant category settings.

System Logs or CEF Example

```
<USERNAME> (<USERID>) - Secret Server Settings Import - Failed to import SAML for the
following reason(s): TicketSystem=Only one ticket system can be default.
(IsDefault);SAML=Identity Provider Id was not found in the database. Check that it was not
modified after export. (IdentityProviderId)
```

SS.log Examples

- `ERROR Delinea.Logging.ILogWriter - <USERNAME> (<USERID>) - Secret Server Settings Import - Failed to import SAML for the following reason(s): TicketSystem=Only one ticket system can be default. (IsDefault);SAML=Identity Provider Id was not found in the database. Check that it was not modified after export. (IdentityProviderId)`Secret Server.
- `ERROR Thycotic.Logging.ILogWriter - <USERNAME> (<USERID>) - Secret Server Settings Import - Failed to import some settings due to the following reason(s): Security=Access Denied;Login=Insufficient permissions to edit Radius settings. (Radius),Insufficient`

```
permissions to edit Delinea One or OpenId settings. (OpenIdConnect),Insufficient
permissions to edit Duo settings. (Duo);TicketSystem=Only one ticket system can be
default. (IsDefault);SAML=Access Denied
```

Errors and Resolutions

Sample Error	Resolution
SAML=Access Denied	Need Administer Configuration SAML permission to update SAML settings.
SAML=Identity Provider Id was not found in the database. Check that it was not modified after export. (IdentityProviderId)	For SAML, the IdentityProviderId provided in the import file was not found in the database. If intending to add a new one, set this to 0.
TicketSystem=Only one ticket system can be default. (IsDefault)	For TicketSystem, IsDefault is set to true when there is already one set to true in the database. If intending to set it to true, set the other one to false.
Insufficient permissions to edit Radius settings. (Radius) Insufficient permissions to edit Duo settings. (Duo)	Need Administer Configuration Two Factor permission to update Radius or Duo settings.
Insufficient permissions to edit Delinea One or OpenId settings. (OpenIdConnect)	Need Administer OpenID Connect permission to update Open ID Connect settings.

Integrations

Note: This list is current as of June 6, 2024. For a comprehensive list of Secret Server integrations see Delinea Integrations Documentation section.

Delinea Integrations

Integrations created by Delinea are code or applications that are not sold by Delinea for monetary compensation. They are provided free of charge for the use of Delinea customers and in some cases are available to the public.

Integrations are supported to the extent of the third-party product procedures documented for those integrations. Contact the third-party for customized setup of the integrated product.

Below is a list of Delinea integrations:

- Google Authenticator Integration with Secret Server
- Google Cloud Platform Discovery Integration with Secret Server
- AWS Discovery Integration with Secret Server
- Authy
- Automation Anywhere
- Atlassian

- Blue Prism
- BMC
- ConnectWise
- DUO Security
- Hardware Security Modules (HSM)
- IBM
- Radius
- JDBC Proxy Driver
- Microsoft
- OneLogin
- Okta
- Palo Alto Networks
- Rapid7
- Red Hat (IBM)
- SAP Hana
- SCIM Connector
- ServiceNow
- Slack
- Splunk
- Terraform
- UiPath Orchestrator

Third-Party Integrations

This category of integration encompasses any code or script which integrates with Delinea usually by API that is written by a third-party vendor. Delinea does not guarantee that third-party code is written correctly or that it respects Delinea product limitations.

For instance, the third-party code may fail to respect Token expiry or issue calls too quickly without waiting for responses and time-outs. Third-party integrations are supported by verifying that the Delinea application is functioning correctly. Delinea does not support, code or maintain third-party code or scripts.

For commercially sold third-party products which have vendor support, Delinea may elect to attend calls. The third-party product must be able to provide a knowledgeable resource and share specifics about how they integrate with the Delinea application. The goal of such calls would be to advise the third-party vendor about what they need to change to better integrate with Delinea products.

> Assistance with configuration or troubleshooting these tools with third-party systems is not within the scope of what the Delinea Support organization can provide at this time. Delinea does not guarantee that every configuration of third-party systems will work with 3rd-Party integrations. Assistance with the use of these tools, configuration, or troubleshooting for customization of Delinea products can be worked on as part of a paid agreement with the Professional Services team.

Below is a list of third-party integrations:

- Argo
- Authomize
- BeyondTrust
- BMC
- Devolutions
- Entrust
- Fortanix
- IBM
- Imprivata
- Keyfactor
- LogicMonitor
- Monokee
- mRemoteNG
- Omada
- PagerDuty
- Qualys
- Royal TS
- Secure ID
- SailPoint
- Saviynt
- SurePassID
- Tenable
- Teradata
- Thales
- Trusona
- Twosense
- UIPath
- WitFoo

Maintenance Mode

Secret Server's maintenance mode is a crucial feature that allows administrators to temporarily prevent users from making changes to roles, secrets, or secret-related data such as dependencies, templates, and password requirements. This mode is particularly useful during critical operations like migrating the Secret Server application to a new server, performing upgrades, or rotating encryption keys to ensure data integrity and prevent corruption. While in maintenance mode, users can still view secrets in a read-only state, but cannot modify them or change their checkout status. Maintenance mode remains active until it is manually disabled, providing a controlled environment for administrators to perform necessary maintenance tasks without risking unauthorized changes or data loss.

Enabling Maintenance Mode

Turning on maintenance mode allows you to temporarily prevent users from changing roles, secrets, or secret-related data such as dependencies, templates, and password requirements. For example, you would want to enable Maintenance Mode while migrating the Secret Server application to a new server with a different domain.

To turn on Maintenance Mode:

1. Go to **Admin > Server Nodes**.
2. In the related node click the edit icon.
3. Check the box next to the Maintenance Mode.
4. Click **Save**.

   Note: When Secret Server is in Maintenance Mode, a notification bar is displayed to alert users.

To return Secret Server from Maintenance Mode to normal operation, click the edit icon again and uncheck the box next to Maintenance Mode. When done, click **Save**.

> When Secret Server is in its normal running mode, the Maintenance Mode notification bar is no longer displayed to users.

Maintenance Mode FAQ

What is Maintenance Mode?

Maintenance mode prevents users from changing secrets or secret-related data such as dependencies, secret templates, and password requirements.

Why do we need Maintenance Mode?

When secret key rotation takes place, or the HSM configuration is changed, Secret Server needs to ensure that no data corruption occurs. To mitigate this, these operations turn on maintenance mode, which puts Secret Server into read-only mode. We also recommend manually enabling maintenance mode before performing upgrades.

Can I still access my Secrets when Maintenance Mode is turned on?

Yes. Secrets will be read-only, but you can still view them, including secrets that are double-locked or protected by "require approval for access." You are unable to change the checkout status of a secret during maintenance mode.

This means if the secret is currently checked-in, you will be unable to check it out. If the secret is currently checked out, it cannot be checked in until the system leaves maintenance mode.

How long does Maintenance Mode last?

Maintenance mode lasts until it is manually disabled.

How do you enable and disable Maintenance Mode?

To enable and disable Maintenance Mode, see "Enabling Maintenance Mode" on the previous page.

Object Metadata

Overview

Object metadata allows you to store extended information on several Secret Server objects including users, groups, folders or secrets via the user interface or REST API. You can store most data types, including strings, Boolean values, numbers and users. You can combine this metadata into sections containing named fields of your defined types.

Unlike pre-existing object fields, this metadata is flexible and dynamic. No coding, structural changes, or database schema changes are required. The only constraint is a role permission that controls who can add metadata fields or sections, which is granular down to the sections level on a given entity. For example, users that can view a user might be allowed to edit the values in the "public" metadata section and users that can edit a user might be allowed to edit all the sections for that user.

Features

Secret Server object metadata features:

- You can store user-defined metadata sections and field values on users, groups, folders, or secrets.
- Sections and fields are defined once and can be used across any applicable object. This allows for a common description across all objects. For example, all the objects could have metadata fields for business owner, source system, and corporate department name.
- Metadata fields are grouped or organized into sections.
- When viewing metadata, only populated fields appear. Field names with blank values are never present.
- You can define who can edit which sections via a role permission and through setting the View or Edit permissions for the object.
- Each object maintains an audit history for all metadata fields, including previous values and who defined them.
- Audit history is viewable as a basic line chart for metadata fields stored as numbers. This provides a historical value table, as opposed to an audit log.

Example Use Cases

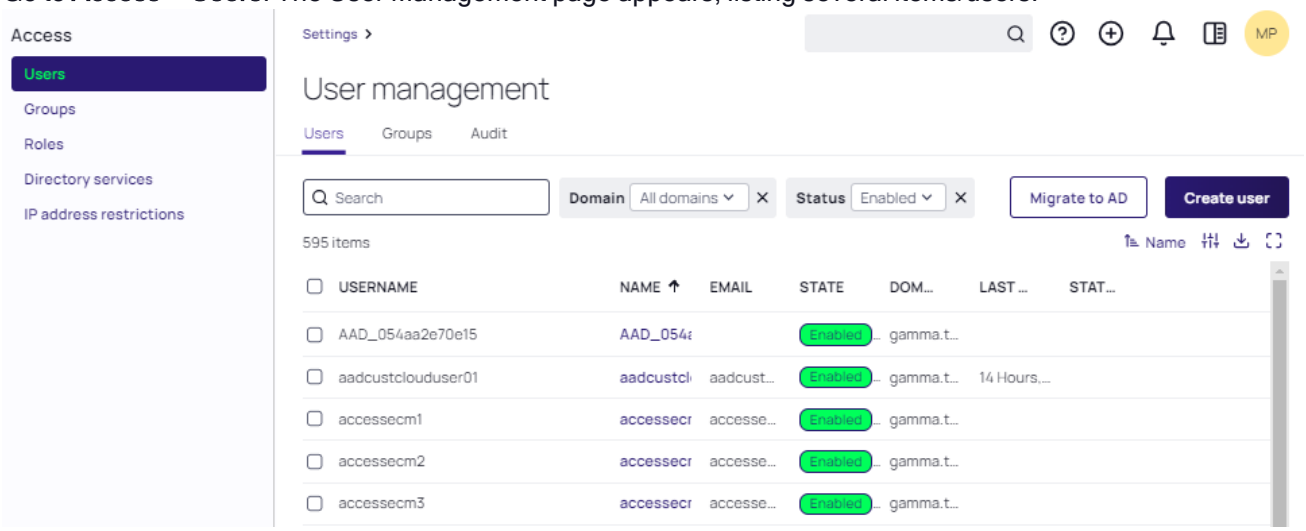
There are many ways to use Secret Server object metadata, such as:

- Defining common attributes from a corporate directory that are not available for a standard user, such as manager, hire date, or department.
- Allowing defined users to add data to an object without allowing that user to edit the object itself. For example, the user can not edit a certain secret but can add notations on the secret that are useful to others accessing the object.
- Storing external system link identifiers for integrations. For example, the employee ID from the HR system could be stored in the metadata for users. Integration jobs could then query this ID from the metadata and use it for synchronization.
- Adding a **department owner** field on a folder to store which department owns it. For instance, the folder contains secrets for marketing as defined by the metadata field.
- Avoiding users putting numerous items into the notes field on a secret, resulting in a disorganized mess. With metadata, those items could be stored in properly named fields. This organizes the notes and allows them to be easily searched without having to parse a block of text.

Adding Object Metadata

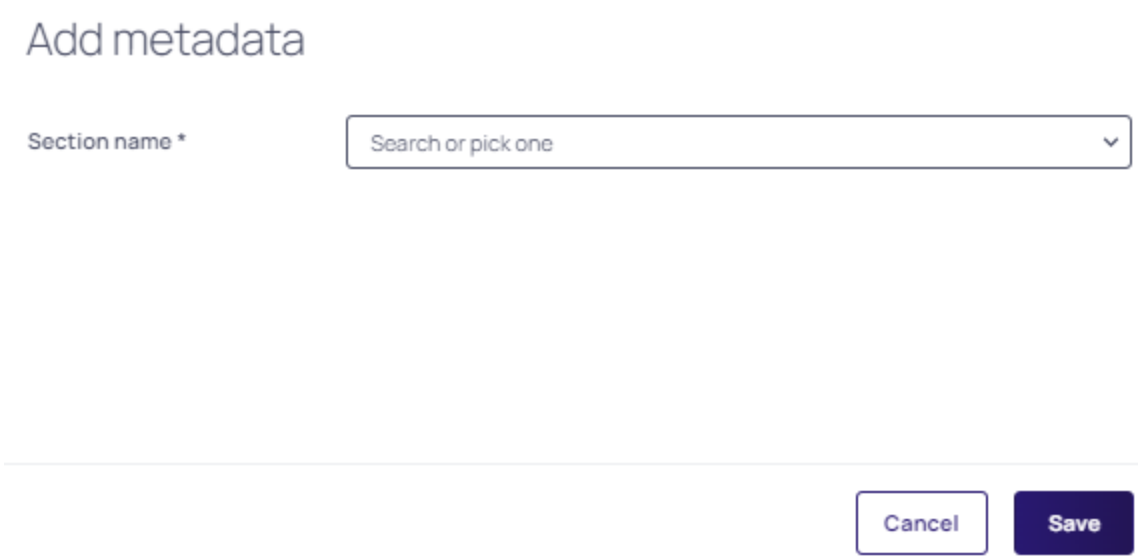
This instruction is on a user object. The process for folders, groups, and secrets is very similar.

1. Go to **Access > Users**. The User Management page appears, listing several items/users:

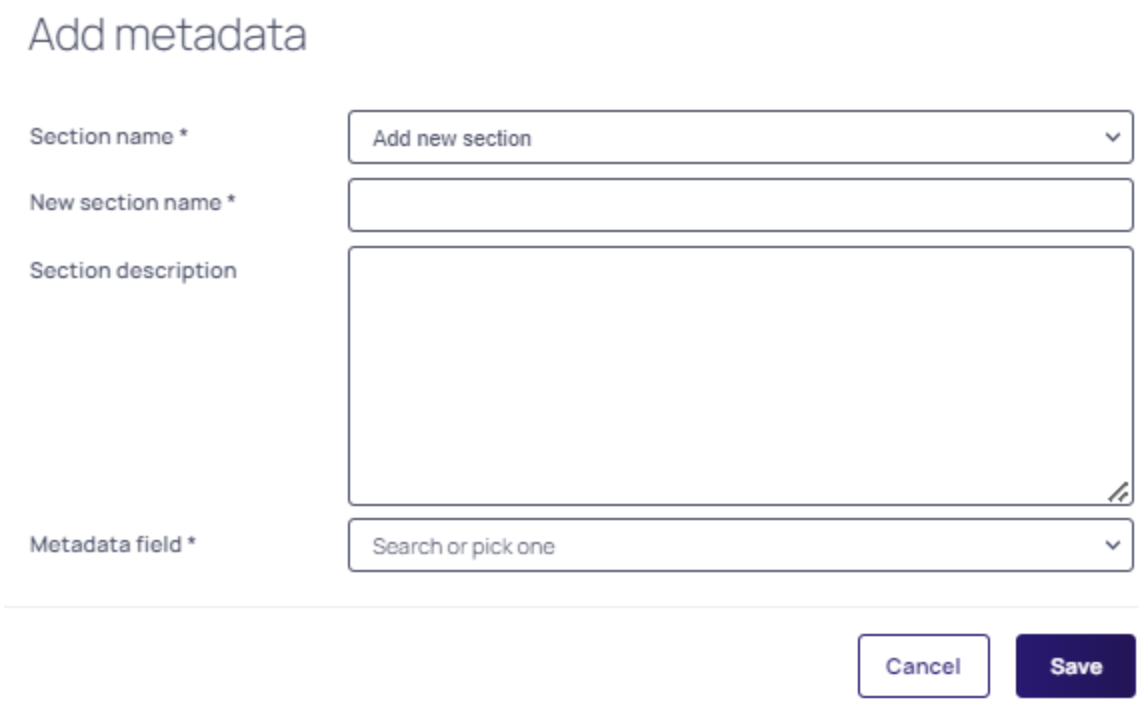


2. Select a user you want to edit. The user's page appears.
3. Select the **Metadata** tab.

4. Click the **Add Metadata** button. The Add Metadata popup appears:



5. Click the **Section Name** dropdown list and select **Add New Section**. Additional controls appear:

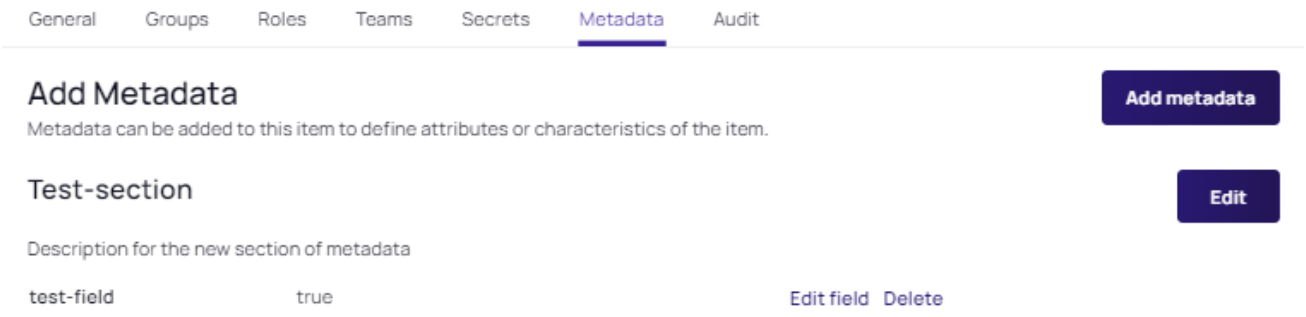


6. Type the section name in the **New Section Name** text box.
7. (Optional) Type a description of the field in the **Section Description** text box.

8. Click the **Metadata Field** dropdown list and select **Add New Field**, more controls appear: **Metadata field name** and **Field type**.
9. Fill in the **Metadata field name** text box.
10. Click the **Field Type** dropdown list and select the desired data type. For our example we chose *Boolean*.
11. (Optional) Select the **Value** check box if you want the field to be pre-populated to true. This only applies to the Boolean data type.

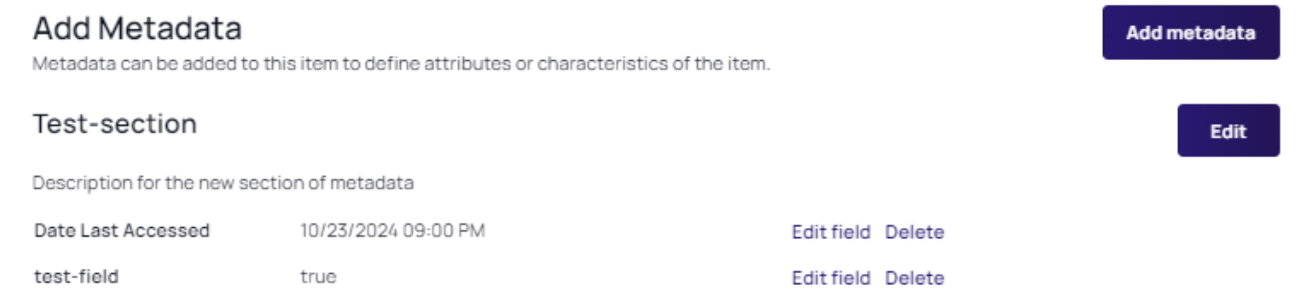
> Boolean fields always appear later because they always have a value. Empty value fields do not appear.

12. Click the **Save** button. The new section and metadata field appears:



The section appears in the top-left, with the field right under it. A true or false value is visible denoting if the user has this test-field.

13. Click the **Add Metadata** button again to add another field to the section. The Add Metadata popup returns.
14. This time, click the **Section Name** dropdown list and select the section that you just created.
15. Click the **Metadata Field** dropdown list box and select **Add New Field**.
16. Type a name for the new field in the **Metadata Field Name** text box.
17. Click the **Field Type** dropdown list to select *Date / Time*. Date and time text boxes appear. Add a date and time, do not leave them empty.
18. Click the **Save** button. The new field appears on the Metadata tab:



19. Add additional fields as required or desired. For our example, this makes the section, the test-field with its Boolean value, and the date / time field available for use across all Secret Server users.

Deleting Object Metadata

1. Go to **Access > Users**. The User Management page appears, listing several items / users:
2. Select a user you want to edit. The user's page appears.
3. Select the **Metadata** tab.
4. Select a section you want to modify fields for and then click **Delete** for the field you no longer need. A prompt appears double checking you want to proceed as deleting a field will also delete its history and cannot be undone.

> Deleting all fields of a section will automatically delete the metadata section as whole. No warning prompt will be issued for this action, so double check before deleting all fields of a section.

Best Practices

How your organization uses object metadata requires some forethought, including:

- Will you allow anyone to add metadata or only a specific set of individuals? This is controlled by applying the above mentioned role.
- How do you want to standardize the naming of sections and fields? One user might call the same field *business owner* and another might call it *subject expert* if you do not establish the field nomenclature up front.
- Do you want to create a "public" field section that is available to all users to edit, even those with read only permission on the object?

Ticket System Integration

Introduction

Secret Server can allow users to enter a ticket number when viewing a secret. This number can be validated through a regular expression, and can also be marked as required, if needed. Secret Server can integrate with third party ticket systems. See below for more information.

Configurable Settings

Navigate to **Admin > Ticket system** - the Ticket systems page opens. Here you can view the list of all the existing Ticket systems. Click on a Ticket system from the list to edit it.

Auditing: The ticket number appears in the audit log and can be queried in reports. If the **View Ticket URL** has been set, the log shows the ticket number as a hyperlink linking to the external ticket system.

Ticket System Settings Section

You can add multiple ticket systems from the Ticket systems section. To add a new system, click **Create Ticket System** at the top right. The New ticket system widow will open. Enter the following settings for your Ticket system settings section:

- **Name**: Enter the related ticket system name.
- **Description**: Enter your ticket system description.
- **Enabled**: Check to enable the ticket system.
- **Default**: Check to make this ticket system the default. If set as default, then this system will be selected by default when a user makes a request. Only one ticket system can be set as default. When saving as default, the previous that was the default will be deselected.
- **Ticket number label**: Enter the desired ticket number label, the text that displays next to the Ticket Number box on the Comment or Request Access page.
- **Ticket number reason options**: This option allows fine-grained control of what the user must enter when Require Comment is enabled and ticket system integration is turned on.
  - **Only Require Reason**: Ticket number is optional, reason is required.
  - **Require Both Ticket Number and Reason**: Ticket number and reason are required.
  - **Ticket Number or Reason Required**: Either ticket number or reason must be entered.
  - **Only Require Ticket Number**: Ticket number is required, reason is optional.
- **Ticket system publicly available**: This configuration is enabled by default, and if it is enabled, the ticketing system integration will attempt to run from the web server rather than the distributed engine. This setting is crucial for environments using Secret Server Cloud, as it may lead to failures if the integration is expected to run from DE. If you use Secret Server Cloud it is recommended to disable this field and select a **Site** from the dropdown that will appear below when the Ticket system publicly available field is disabled.
- **Site**: Select the related site from the dropdown to run for your ticket system when the ticket system publicly available field is disabled.

Ticket System Integration Section

Enter the following settings for your Ticket system integration section (also see Third Party Integrations for related details):

- **Type**: Select the related ticket system from the dropdown:
  - **Ticket number validation**: Secret Server can require users to enter a ticket number when viewing a secret. Admins can track access to secrets based on an external ticket system. After the ticket system is enabled in Secret Server, a user can enter a ticket number on the Comment screen or the Request Access screen. The secret needs to have Require Comment or Requires Approval for Access enabled to allow the user to enter a ticket number. When a ticket number is required, this secret setting is displayed as "Require Comment/Ticket Number" on the Security tab.
  - **BMC Remedy Incident Management**: See BMC Remedy Integration for more details.
  - **BMC Remedy Change Management**: See BMC Remedy Integration for more details.
  - **Custom Ticketing System (PowerShell)**: See Atlassian JIRA Integration (PowerShell), ManageEngine ServiceDesk Plus Integration (PowerShell), and PowerShell Ticketing Integration for more details.

  - **ServiceNow Change Management**: See ServiceNow Integration for more details.
  - **ServiceNow Incident Management**: See ServiceNow Integration for more details.

- **View Ticket URL Template:** The format of the URL to be used for viewing the ticket. This is placed in the audit log so you can easily view the corresponding ticket from Secret Server. The Ticket URL Format field can be edited on the "Ticketing System Integration" tab of the configuration page. In this field, the $TICKETID parameter will be replaced by the ticket number that is entered by the user. For example, if you specify the template as `http://myticketingsystem/ticket.aspx?ticketid=$TICKETID`, and a user enters 5125-242 as the ticket number, a link will appear in the audit log to `http://myticketingsystem/ticket.aspx?ticketid=5125-242`.
- **Ticket Number Format Pattern (Regex):** A regular expression to use for validating the ticket number entered. This can help prevent typos in the number. Considerations:
  - If you are using ticketing system integration, you can set a ticket pattern on the Ticket System Integration tab of the Configuration page.
  - If you do not want to restrict what ticket numbers a user can enter, you can leave the Ticket Number Validation Pattern (Regex) text box empty.
  - If you do want to restrict it, you can enter a regular expression in the text box. The ticket number entered must match the regular expression.
  - If you are supported and need assistance setting up a validation pattern, feel free to email support@delinea.com.
  - Here is an example for a ticket pattern that must be a valid number: `^[0-9]+$`
- **Ticket Number Validation Error Message:** The error message to display to the user when their entered ticket number fails the validation pattern regex.

Ticket Number Override Section

Here, you can designate approvers to manually approve tickets if ticket validation fails from something like an external system error. The settings are:

- **Enable Override:**
  - Yes, always bypass ticket validation and require approval
  - Off (do nothing upon failure)
- **Secret Owners Can Approve Override check box:** When the box is selected, an owner can enter a ticket number and request access when they are notified about a validation failure. The request is emailed to the approvers. An approver can click on a link in the email that takes them to the Ticket Overrides section of the Secret Server Inbox where they can approve or deny the request. Once the request is approved, the requester can check out the secret from the same dialog box where they requested the approval.
- **Override Approvers:** A list of designated approvers for overriding validation errors. Approvers can be added with the Add Approver link.

Third-Party Integrations

Secret Server can integrate into third-party ticket systems as well. Those supported are listed below. You can add multiple ticket systems to Secret Server by clicking New Ticket System. You can make a specific ticket system the

default system used by Secret Server by clicking the System link and then clicking the Set as Default button.

The third-party integrations:

- "Atlassian JIRA Integration (PowerShell)" below
- "BMC Remedy Integration" on page 228
- "ManageEngine ServiceDesk Plus Integration (PowerShell)" on page 231
- "PowerShell Ticketing Integration" on page 235
- "ServiceNow Integration" on page 233

Atlassian JIRA Integration (PowerShell)

Secret Server can integrate with Atlassian JIRA via PowerShell. This integration includes validating ticket numbers and their status, and adding comments.

For more information about integrating ticket systems with PowerShell, see "PowerShell Ticketing Integration" on page 235.

Requirements

- PowerShell, see "Creating and Using PowerShell Scripts" on page 1475.
- "Configuring CredSSP for WinRM with PowerShell" on page 1479.

> Atlassian has deprecated TLS 1.0 and 1.1, and will support only TLS 1.2 and 1.3 going forward. See Deprecating TLSv1 and TLSv1.1 for Atlassian Cloud Products.

Ticket Number Validation Pattern (Regex)

Before making a call to the ticket validation script, you can have Secret Server validate that the number matches a pattern. For example, you will probably have multiple JIRA projects and will want your users to specify these correctly. One regex would be to simply allow any project name followed by a dash and numbers:

```
^\w{3}-\d+$
```

Or perhaps you want to specifically match projects that you know are real followed by a dash and numbers:

```
^((PRO1)|(PRO2))-\d+$
```

For more information, see **Setting a Ticket Pattern Regex** on the "Ticket System Integration" on page 222 page.

Validating Ticket Status

> The system credentials are specific to your ticketing system. You can use any secret using the username and password extended mapping as your system credential. You can add other arguments in the secret's fields and reference them in your script.

You need to create a PowerShell script to retrieve and validate tickets. This integration assumes that the user will pass in the full ticket name, including the project name. For example: `(PROJ-123)`. This could easily be extended so that multiple JIRA instances could be made for each project. In that case, you could have the user provide only the

ticket number and pass in an argument to the script that specifies the project. This implementation also assumes that any ticket not in "Closed" status is invalid.

```
$ticket = $args[0]
$user = $args[1]
$password = $args[2]
$url = $args[3]
$closedStatus = "Closed"
$fields = "status"
$p = $password | ConvertTo-SecureString -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PsCredential($user,$p)
$getStatusMethod = "$url/rest/api/latest/issue/$ticket"
function ConvertTo-UnsecureString([System.Security.SecureString][parameter
(mandatory=$true)]$SecurePassword)
{
    $unmanagedString = [System.IntPtr]::Zero;
    try
    {
        $unmanagedString =
[Runtime.InteropServices.Marshal]::SecureStringToGlobalAllocUnicode($SecurePassword)
        return [Runtime.InteropServices.Marshal]::PtrToStringUni($unmanagedString)
    }
    finally
    {
        [Runtime.InteropServices.Marshal]::ZeroFreeGlobalAllocUnicode($unmanagedString)
    }
}
function ConvertTo-Base64($string)
{
   $bytes  = [System.Text.Encoding]::UTF8.GetBytes($string);
   $encoded = [System.Convert]::ToBase64String($bytes);
   return $encoded;
}
function ConvertFrom-Base64($string)
{
   $bytes  = [System.Convert]::FromBase64String($string);
   $decoded = [System.Text.Encoding]::UTF8.GetString($bytes);
   return $decoded;
}
function Get-HttpBasicHeader($Credentials, $Headers = @{})
{
 $b64 = ConvertTo-Base64 "$($Credentials.UserName):$(ConvertTo-UnsecureString
$Credentials.Password)"
 $Headers["Authorization"] = "Basic $b64"
 return $Headers
}
try
{
    $headers = Get-HttpBasicHeader $credentials
    $response = Invoke-RestMethod -Method Get -uri $getStatusMethod -Headers $headers -
ContentType 'application/json'
    if($response.fields.status.name -eq $closedStatus)
    {
```

```
        throw "JIRA ticket ($ticket) is closed."
    }
}
catch
{
    $exception = $_.Exception
    if ($exception.Response.StatusCode.value__ -eq 404)
    {
        throw "JIRA ticket ($ticket) does not exist."
    }
}
```

Adding Comments to Tickets

To add comments to tickets, you will need to create the script below.

```
$ticket = $args[0]
$comment = $args[1]
$user = $args[2]
$password = $args[3]
$url = $args[4]
$p = $password | ConvertTo-SecureString -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PsCredential($user,$p)
function ConvertTo-UnsecureString([System.Security.SecureString][parameter
(mandatory=$true)]$SecurePassword)
{
    $unmanagedString = [System.IntPtr]::Zero;
    try
    {
        $unmanagedString =
[Runtime.InteropServices.Marshal]::SecureStringToGlobalAllocUnicode($SecurePassword)
        return [Runtime.InteropServices.Marshal]::PtrToStringUni($unmanagedString)
    }
    finally
    {
        [Runtime.InteropServices.Marshal]::ZeroFreeGlobalAllocUnicode($unmanagedString)
    }
}
function ConvertTo-Base64($string)
{
   $bytes  = [System.Text.Encoding]::UTF8.GetBytes($string);
   $encoded = [System.Convert]::ToBase64String($bytes);
   return $encoded;
}
function ConvertFrom-Base64($string)
{
   $bytes  = [System.Convert]::FromBase64String($string);
   $decoded = [System.Text.Encoding]::UTF8.GetString($bytes);
   return $decoded;
}
function Get-HttpBasicHeader($Credentials, $Headers = @{})
```

```
{
 $b64 = ConvertTo-Base64 "$($Credentials.UserName):$(ConvertTo-UnsecureString
$Credentials.Password)"
 $Headers["Authorization"] = "Basic $b64"
 return $Headers
}
try
{
    $updateObject = @{'body'=$comment}
    $body = $updateObject | ConvertTo-Json
    $addComment = "$url/rest/api/latest/issue/$ticket/comment"
    $headers = Get-HttpBasicHeader $credentials
    $response = Invoke-RestMethod -uri $addComment -Headers $headers -Method Post -
ContentType "application/json" -Body $body
    if ($response.body -ne $comment)
    {
        throw "There was an issue adding a comment to the ticket ($ticket)."
    }
}
catch
{
    $exception = $_.Exception
    if ($exception.Response.StatusCode.value__ -eq 404)
    {
        throw "The ticket ($ticket) does not exist.";
    }
    if ($exception.Response.StatusCode.value__ -eq 400)
    {
        throw "There was an issue adding a comment to the ticket ($ticket)."
    }
    throw "There was an unhandled issue with adding a comment: " + $exception.ToString()
}
```

BMC Remedy Integration

Overview

Secret Server can integrate with BMC Remedy's Incident and Change Management. This integration includes validating ticket numbers, their status, and adding work detail items to the request.

The integration with BMC Remedy leverages the out-of-the-box, SOAP-based Web services that are installed with the ITSM product installation. These services must be installed on your mid-tier BMC Remedy server to allow for this integration if they are not already installed and configured.

Requirements

- BMC Remedy SOAP Web Services enabled
- A username and password that has access to execute the Web services. This can be set up in the developer studio by accessing the application in the navigator and viewing Permissions for the `CHG_ChangeInterface_WS` or `HPD_IncidentInterface_WS`. This user should also have access to query requests and add work items to requests for the appropriate module.

- Secret Server environment needs to be able to connect to the BMC Remedy Web services via port 80 or 443. SSL is highly recommended because the SOAP messages contain a username and password.

Configurable Settings

Validating Ticket Status

When a BMC Remedy request number is entered into Secret Server, the status of that request is retrieved to ensure that it is an open state. For example, if an incident number is entered that is in the "Closed" state, the user is informed that the ticket is closed.

Incident Management: Service Incident request cannot be closed or canceled. Change Management: Change management requests cannot be complete, closed, or canceled.

View Ticket URL Template

The format of the URL to be used for viewing the ticket. This is placed in the audit log so you can easily view the corresponding ticket from Secret Server. Depending on your version of BMC Remedy, the URL to link directly to a request may be slightly different.

Incident management:

```
https://<midtier_server>//arsys/forms/<server
name>/SHR%3ALandingConsole/Default+AdmSearchTicketWithQual&F304255610='Incident
Number'%3D%22$TICKETID%22
```

Change management:

```
https://<midtier_server>//arsys/forms/<server
name>/SHR%3ALandingConsole/Default+AdmSearchTicketWithQual&F304255610='Change
Number'%3D%22$TICKETID%22
```

Ticket Number Format Pattern (Regex)

Before even making a call to the BMC Remedy Web service, you can have Secret Server validate that the number matches a pattern. For example, your incident numbers might all be prefixed with "INC" and you want to ensure users enter the prefix. Some sample expressions to validate the ticket number are listed below:

Incident management: `^INC_CAL_[\d]{7}$`

Change management: `^CRQ_CAL_[\d]{7}$`

Ticket Number Validation Error Message

The error message to display to the user when their entered ticket number fails the validation pattern regex.

Service Endpoint URL

This is the URL for the SOAP-based Web services. Below are some samples for what is expected. You can find the actual endpoint using BMC Remedy Developer Studio and accessing the correct application from the AR System Navigator and viewing the Web services section of the application.

Incident management: `HPD_IncidentInterface_WS`

Change management: `CHG_ChangeInterface_WS`

System Credentials

Select or create a secret that contains the username and password for a user that has access to execute the SOAP Web services. The username and password are added to the authentication header for the SOAP request.

> The system credentials are specific to your ticketing system. You can use any secret using the username and password extended mapping as your system credential. You can add other arguments in the secret's fields and reference them in your script.

Authentication

If your installation of BMC Remedy uses an authentication server, enter it in this text-entry field. Most installations allow this text-entry field to be blank.

Add Comments to Ticket

Check this box if you want the comment that a user enters to be added to the request in BMC Remedy. This adds information such as the secret for which access is requested, who requested access, and the requester's comments.

Comment Work Type

When a comment is added to a request as a work item, the Work Item type is required. "General Information" is selected by default, but all default Work Type options are supported.

Testing Your Integration Setup

After configuring the ticket system (see configurable settings below), use the **Test Validation** button to verify that Secret Server can successfully access BMC Remedy. This button opens a dialog in which you can enter a ticket number from BMC Remedy. This validation process returns success or an error code. BMC Remedy may not return much detail in the error message so you need to look at the BMC Remedy API log to see a detailed error message, see the next section.

BMC Remedy Error Messages

When Secret Server calls the BMC Remedy SOAP-based web services, there are times that BMC will only return a 500 error without any details of the exception. You can see the details of this exception from the BMC Remedy server logs as described below:

1. Log on BMC Remedy as a user with access to the administrative console.
2. Navigate to **AR System Administration** from the main menu.
3. Navigate to **System > General > Service Information**.
4. Click the **Log Files** tab.
5. Click to enable the **API Log** check box.
6. Click the **Apply** button.

7. Once enabled, you can click **View** from this window to see the log or navigate to the mid-tier server's file system at the location specified. Details of the SOAP web service exception are written to the log file including a stack trace.

ManageEngine ServiceDesk Plus Integration (PowerShell)

Secret Server can integrate with ManageEngine ServiceDesk Plus via PowerShell. This integration includes validating ticket numbers and their status, and adding comments (referred to as notes in ServiceDesk Plus).

For more information about integrating ticket systems with PowerShell, see "PowerShell Ticketing Integration" on page 235.

Requirements

- PowerShell, see "Creating and Using PowerShell Scripts" on page 1475.
- Access to the REST API for your ManageEngine ServiceDesk Plus instance.
- "Configuring CredSSP for WinRM with PowerShell" on page 1479.

Ticket Number Validation Pattern (Regex)

Before making a call to the ticket validation script, you can have Secret Server validate that the number matches a pattern. For more information, see **Setting a Ticket Pattern Regex** on the "Ticket System Integration" on page 222 page.

Validating Ticket Status

> The system credentials are specific to your ticketing system. You can use any secret using the username and password extended mapping as your system credential. You can add other arguments in the secret's fields and reference them in your script.

You need to create a PowerShell script to retrieve and validate tickets. This integration assumes that the administrator will set the technician key for accessing ServiceDesk Plus in the script. This could easily be extended to pass in the key as an argument so that it can be managed from the ticket system interface.

```
$ticket = $args[0]
$user = $args[1]
$password = $args[2]
$url = $args[3]
$validStatus = "Open"
$fields = "status"
$technicianKey = "<YOUR API GUID>"
$p = $password | ConvertTo-SecureString -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PsCredential($user,$p)
$getStatusMethod = "$url/sdpapi/request/$ticket"
try
{
    $postParams = @{OPERATION_NAME='GET_REQUEST';TECHNICIAN_KEY=$technicianKey}
```

```
    [xml]$response = Invoke-WebRequest -Uri $getStatusMethod -Method POST -Body
$postParams -Credential $credentials
    if ($response.API.response.operation.result.status -ne "Success")
    {
     throw "Response not successful." + $response.API.response.operation.result.message
    }
    $statusNode = $response.API.response.operation.Details.ChildNodes | Where-Object {
($_.name -eq "status") }
    if($statusNode.value -ne $validStatus)
    {
        throw "Manage Engine Service Desk Plus ticket ($ticket) is not in Open status."
    }
}
catch
{
    if ($response.operation.result.message -eq "Invalid requestID in given URL")
    {
        throw "Manage Engine Service Desk Plus ticket ($ticket) does not exist."
    }
    throw "Manage Engine Service Desk Plus encounderted an error: " +
$response.operation.result.message
}
```

Adding Comments (Notes) to Tickets

To add comments (notes) to tickets you will need to create the script below. Other considerations are to pass in whether or not the note should be public.

```
$ticket = $args[0]
$comment = $args[1]
$user = $args[2]
$password = $args[3]
$url = $args[4]
$technicianKey = "<YOUR API GUID>"
$isPublic = 'true'
$p = $password | ConvertTo-SecureString -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PsCredential($user,$p)
# Clean comment input for use in XML
$comment = $comment.replace('&', '&amp;').replace("'", ''').replace('"', '"').replace('<',
'<').replace('>', '>')
$inputData = @"
<Operation>
    <Details>
        <Notes>
            <Note>
                <isPublic>$isPublic</isPublic>
                <notesText>$comment</notesText>
            </Note>
        </Notes>
    </Details>
```

```
</Operation>
"@
$URI = "$url/sdpapi/request/$ticket/notes"
$postParams = @{OPERATION_NAME='ADD_NOTE';TECHNICIAN_KEY=$technicianKey;INPUT_
DATA=$inputData}
$response = Invoke-WebRequest -Uri $URI -Method POST -Body $postParams
[xml]$responseContent = $response.Content
if ($responseContent.API.response.operation.result.status -ne "Success")
{
    $message = $responseContent.API.response.operation.result.message;
    throw "Unable to add comment to ticket ($ticket). Message: " + $message
}
```

ServiceNow Integration

Introduction

Secret Server can integrate with ServiceNow's Incident and Change Management service. This integration includes validating ticket numbers, checking their status, and adding Work Detail items to the request. The integration with ServiceNow leverages out-of-the-box REST-based Web services.

Requirements

- A ServiceNow instance running the Eureka version or later with REST services enabled.
- A username and password that have access to execute the REST services, specifically GET and MODIFY on the following tables: **Change Request** and **Incident**.
- The Secret Server environment needs to be able to connect to the ServiceNow Web services via port 80 or 443. SSL is highly recommended because the REST messages authenticate with a username and password.

Configurable Settings

The ServiceNow Integration has several configurable settings that the user needs to explore:

- **View Ticket URL Template**:

  The format of the URL used for viewing the ticket. This appears in the audit log so you can easily view the corresponding ticket from Secret Server.

  Use the following templates for incident or change management instances:

  - Incident management: `https://<instance name>.service-now.com/nav_to.do?uri=incident.do?sysparm_query=number=$TICKETID`
  - Change management: `https://<instance name>.service-now.com/nav_to.do?uri=change_request.do?sysparm_query=number=$TICKETID`

  
  This field specifies the URL that will be used when displaying a link to a ticket in the audit log. In this field, the $TICKETID parameter will be replaced by the ticket number that is entered by the user.

For example, if you specify the **View Ticket URL Template** as `http://myticketingsystem/ticket.aspx?ticketid=$TICKETID`, and Bob enters `5125-242` as the ticket number, a link will appear in the audit log to `http://myticketingsystem/ticket.aspx?ticketid=5125-242`.

- **Ticket Number Format Pattern (Regex)**:

  Before even making a call to the ServiceNow Web service you can have Secret Server validate the number matches a pattern. For example, your incident numbers might all be prefixed with "INC" and you want to ensure they enter this prefix. Some sample expressions to validate the ticket number are listed below:

  - Incident management: `^INC[\d]{7}$`
  - Change management: `^CHG[\d]{7}$`

- **Ticket Number Validation Error Message**:

  The error message to display to the user when their entered ticket number fails the validation pattern regex.

- **Domain Name**:

  This is the domain name of your instance in the following format: `<instance name>.service-now.com`. For example: `dev5859.service-now.com`.

- **System Credentials**:

  Select or create a secret that contains the username and password for a user that has access to execute the REST Web services. Secret Server uses these credentials to authenticate to ServiceNow.

- **Allowed Statuses**:

  The allowed statuses for ServiceNow tickets can vary based on the specific configuration and customization of the platform within an organization. Generally, however, the statuses for each type of ticket are as follows:

  - **Incident Statuses**:
    - **New**: The incident has been logged but not yet reviewed.
    - **In Progress**: The incident is being actively worked on.
    - **On Hold**: Work on the incident is paused, possibly waiting for more information or a third-party response.
    - **Resolved**: A solution has been implemented, and the incident is awaiting confirmation from the user.
    - **Closed**: The incident is confirmed resolved and closed.
  - **Request Statuses"**
    - **New**: The request has been submitted but not yet reviewed.
    - **In Progress**: The request is being fulfilled.
    - **On Hold**: The request is paused, possibly waiting for more information or approval
    - **Completed**: The request has been fulfilled.
    - **Closed**: The request is confirmed completed and closed.

  These statuses can be customized, so it's best to check with your organization's ServiceNow administrator for the exact statuses used.

The comma separated list of statuses provided will be considered legitimate by secret-server.

- **Add Comments to Ticket**:

  Check this box if you want the comment that a user enters to be added to the request in ServiceNow. This adds information such as the Secret to which access is requested, who requested access, and their comments. The comment is added as a work note in the activity section of the request.

Testing your Integration Setup

After configuring the ticket system, use the **Test Validation** button to verify that Secret Server can successfully access ServiceNow. This button opens a dialog in which you can enter a ticket number from ServiceNow. This validation process either succeeds or shows an error code.

Please note, Secret Server validates the ticket number but does not include validation of a ticket's status based on the *Action Status* in the configuration. To validate the status of the ticket customers need to write a script to validate the status based on the code base. Secret Server only validates based on Ticket ID and whether or not SNOW returns an error.

PowerShell Ticketing Integration

Secret Server can integrate with your ticketing system via PowerShell. This integration includes validating ticket numbers, their status, and adding comments. In our example we are connecting to a ServiceNow instance.

See "Creating and Using PowerShell Scripts" on page 1475.

Configurable Settings

View Ticket URL Template

You can configure the view ticket URL if you have a web based ticketing system to allow easy access to link to your ticketing system from Secret Server.

Ticket Number Validation Pattern (Regex)

Before making a call to the PowerShell script you can have Secret Server validate the number matches a pattern. For example, your incident numbers might all be prefixed with "INC" and you want to ensure they enter this prefix. See "Ticket System Integration" on page 222).

Ticket Number Validation Error Message

The error message to display to the user when their entered ticket number fails the validation pattern Regex.

The PowerShell RunAs Credentials

In Secret Sever a domain credential is required to execute the PowerShell script. This is a required field.

System Credentials

The system credentials are specific to your ticketing system. You can use any secret using the username and password extended mapping as your system credential. You can add other arguments in the secret's fields and reference them in your script.

Validating Ticket Status

Overview

To validate tickets you will need to create a PowerShell script to retrieve and validate the ticket. The integration will use arguments to pass custom values to your script. By default we will map certain fields to the first set of arguments. The ticket number will be collected by user input and assigned to the first parameter. When you have your ticketing system credentials mapped to a secret and assigned to the "System Credentials" field in the ticketing system setup, Secret Server inserts UserName and Password as the second and third parameters.

Therefore, for the sample script below, the Ticket Status Script Arguments text box should be only contain`$url` (which is also retrieved from the System Credentials secret), as `$ticket`, `$user` and `$password` are supplied automatically by the system.

Sample Script

```
$ticket = $args[0]
$user = $args[1]
$password = $args[2]
$url = $args[3]
$validStatus = "2"
$fields = "state"
$p = $password | ConvertTo-SecureString -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PsCredential($user,$p)
$getStatusMethod = "$url/api/now/table/incident?sysparm_limit=10&sysparm_
query=number=$ticket&sysparm_display_value=&sysparm_fields=$fields"
$response = Invoke-RestMethod $getStatusMethod -Method Get -ContentType
'application/json' -Credential $credentials
if($response.result.state -ne $validStatus)
{
  throw "Invalid State"
}
```

Adding Comments to Tickets

To add a comment to tickets, create another script to do so. Example:

```
$ticket = $args[0]
$comment = $args[1]
$user = $args[2]
$password = $args[3]
$url = $args[4]
$p = $password | ConvertTo-SecureString -AsPlainText -Force
```

```
$credentials = New-Object System.Management.Automation.PsCredential($user,$p)
$restEndpoint = "$url/api/now/table/incident?sysparm_limit=10&sysparm_
query=number=$ticket&sysparm_display_value=&sysparm_fields=sys_id"
$response = Invoke-RestMethod $restEndpoint -Method Get -ContentType 'application/json' -
Credential $credentials
$id = $response.result.sys_id
$updateObject = @{'work_notes'=$comment}
$body = $updateObject | ConvertTo-Json
$addComment = "$url/api/now/table/incident/$id"
$response = Invoke-RestMethod $addComment -Method Put -ContentType 'application/json' -
Credential $credentials -Body $body
```

Adding Comments to a General Audit Log

In addition to adding comments to specific tickets, you may want general audit entries made in your ticket system. The arguments are passed in the following order.

```
$comment = $args[1]
$user = $args[2]
$password = $args[3]
## custom script here
```

Troubleshooting and Notices

The Troubleshooting section is designed to assist users in diagnosing and resolving common issues they may encounter. It includes a variety of subcategories, such as enabling debug in system logs, identifying release version numbers, addressing HTTP errors, and resolving SSH issues. Additionally, there are topics dedicated to IIS-specific troubleshooting, notices regarding system updates or changes, and other general troubleshooting topics.

> This section is a work in progress and supplements knowledge articles provided by Support. It does **not** contain a complete set of Secret Server troubleshooting and workaround articles.

IIS

> This topic only applies to **Secret Server On-Premises**.

This section is dedicated to troubleshooting issues specific to IIS, Microsoft's web server platform. These resources are designed to help users optimize the performance and functionality of their IIS setups.

Application Pool Load User Profile Setting Must Be Enabled

> This topic only applies to **Secret Server On-Premises**.

Secret Server requires the application pool to have the "load user profile" setting enabled. Secret Server will report a critical alert to notify admins if this setting is not enabled.

> The site will load to give access to secrets but many internal operations will not function correctly so we recommend fixing the issue as soon as possible.

> This applies to version 10.2 and later.

Steps to enable the "load user profile" setting:

1. On each Web server that is running Secret Sever, open IIS Manager.
2. Under the **Application Pool** node on the left, select **Secret Server**.
3. On the right-hand panel, select **Advanced Settings** to get to the full properties.
4. Scroll to the **Load User Profile** setting in the **Process Model** section.
5. Set **Load User Profile** to **True**.
6. Click the **OK** button.
7. Preform an `iisreset` on the server:
   a. Open a Windows command prompt as an administrator.
   b. Type `iisreset`.
   c. Press the **<Enter>** key.

Changing IIS to Not Stop Worker Process in IIS 7.0 and Later

> This topic only applies to **Secret Server On-Premises**.

Overview

When using IIS version 7.0 and above, by default, the worker process terminates after a period of inactivity. If Secret Server is in its own application pool, the application pool will stop after a period of no requests. To make sure that the application pool associated with Secret Server does not stop when idle:

- Set the idle time-out to 0 minutes.
- Set the regular time interval to 0.
- Ensure there are no specific times scheduled for recycling.

Additionally, by default, IIS launches a worker process when the first request for the Web application is received. So if the Secret Server application takes a long time to start, we recommend launching the worker process as soon as IIS is started by setting the start mode to AlwaysRunning to launch the worker process for the Secret Server application pool as soon as IIS is started.

Procedure

To change IIS advanced settings:

1. Open **Internet Information Server (IIS) Manager**: On the taskbar, click **Server Manager**, click **Tools**, and then click **Internet Information Services (IIS) Manager**.
2. In the **Connections** pane, expand the server name.
3. Click **Application Pools**.
4. Locate the application pool Secret Server is running as. To determine this:
   a. Expand **Sites** at the left, then find the website Secret Server is running on.
   b. Click on the Secret Server website or virtual directory (if it is running on one).
   c. Click **Basic Settings** on the right panel. This indicates Secret Server's application pool.
5. Right-click the application pool, and select **Advanced Settings**. The Advanced Settings panel appears.
6. Go to the **(General)** section.
7. Set **Start Mode** to **AlwaysRunning**.
8. Set **Maximum Worker Processes** to **1**.
9. Go to the **Process Model** section.
10. Set **Idle Time-out (minutes)** to **0**.
11. Go to the **Recycling** section.
12. Set the **Regular Time Interval (minutes)** to **0**.
13. Select **Specific Times**.
14. **Either** click the **>** expander arrow to see if there is time specified below. **Or** click the **...** to see if there are any values in the **TimeSpan Collection Editor** dialog box. If so, clear it out.
15. Click the **OK** button. The dialog closes.
16. Click the **OK** button.

Notices

The Security and Technical Notices section serves as a central hub for important updates and alerts regarding system security and technical changes. This area provides critical information about security vulnerabilities, patches, and best practices to safeguard systems from potential threats. Additionally, it includes technical notices that inform users about upcoming changes, software updates, and system enhancements.

Notice: jQuery CVE-2019-11358

Relevance

This Delinea**technical issue** knowledge base article is relevant to:

- Product(s): Secret Server using jQuery 3.2.1
- Version(s): 10.7
- Edition(s): All

Technical Issue

Secret Server 10.7 uses jQuery 3.2.1, which is listed as vulnerable to the jQuery CVE-2019-11358 security issue on the Common Vulnerabilities and Exposures (CVE) list.

Resolution

Delinea removed the jQuery vulnerability from Secret Server's copy of jQuery v3.2.1 by applying a patch (see Related Articles and Resources).

To verify the fix:

1. Navigate to `https://<your_secret_server_URL>/assets/libs/jquery-3.2.1.js`
2. Open the file in a text editor.
3. Search for the string `proto` in the code: ...

```
ERROR: Invalid Code Highlighting Language
```

4. If the string appears, the patch has been applied.

Related Articles and Resources

- NIST website for CVE-2019-11358
- GitHub commits on the fix

> The commit shows two files, the top file is the security fix, and the bottom file is a unit test for the fix. Secret Server does not ship with any jQuery unit tests as found in that second file.

- Common Vulnerabilities and Exposures (CVE) list

Notice: jQuery CVE-2020-11022

Relevance

This Delinea **technical issue** knowledge base article is relevant to:

- Product(s): Secret Server using jQuery 3.2.1
- Version(s): 10.8.000004
- Edition(s): All

Technical Issue

Secret Server 10.8.000004 uses jQuery 3.2.1, which is listed as vulnerable to the jQuery CVE-2020-11022 security issue on the Common Vulnerabilities and Exposures (CVE) list.

Resolution

Delinea removed the jQuery vulnerability from Secret Server's copy of jQuery v3.2.1 by applying a patch (see Related Articles and Resources).

To verify the fix:

1. Navigate to `https://<your_secret_server_URL>/assets/libs/jquery-3.2.1.js`
2. Open the file in a text editor.
3. Search for the string `htmlPrefilter` in the code (line 5919):

```
jQuery.extend( {
htmlPrefilter: function(html) {return html;}
```

4. If the string appears, the patch has been applied.

Related Articles and Resources

- NIST website for CVE-2020-11022
- GitHub commits on the fix

> The commit shows multiple files, the top file is the security fix, and the bottom files are unit tests for the fix. Secret Server does not ship with any jQuery unit tests.

- Common Vulnerabilities and Exposures (CVE) list

Security Advisory 2019

Detection

During a security review of Secret Server on June 4, 2019, an internal security team found the security issue described below. The issue was also detected later by an internal team in Password Reset Server.

The Security Issue

An attacker with administrator permissions could modify the input field data in one specific location to execute a SQL injection attack against Secret Server or Password Reset Server. This means that the attacker could append, modify or delete data in the Secret Server or Password Reset Server SQL databases, and upgrade their access to code execution on the SQL server.

Common Vulnerability Scoring System Version 3.0

The CVSS score for this issue is 9.1. The CVSS vector string is CVSS:3.0/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H

Products Affected

- Secret Server On-Premises version 10.6.000026 and earlier.
- Secret Server Cloud, which was updated June 15, 2019 to permanently remove this issue for all customers.

- Password Reset Server earlier than version 5.1.000005.

Recommended Actions

- All Secret Server On-Premises customers should upgrade to version 10.6.000027 or later.
- All Password Reset Server customers should upgrade to version 5.1.000005 or later.
- Secret Server Cloud users do not need to take any action.

HTTP Errors

This subsection focuses on diagnosing and resolving specific HTTP-related issues. Within this section, you will find folders dedicated to troubleshooting particular HTTP error codes, such as the "HTTP 404.2 Error" related to ISAPI and CGI restrictions and the "HTTP 404.17 Error" associated with the .NET Framework.

HTTP Error 404.17 - Not Found After Upgrading .NET Framework Version

Error

After upgrading Secret Server and changing the CLR version, when attempting to load Secret Server, you receive the following error in Internet Explorer:

```
HTTP Error 404.17 - Not Found The requested content appears to be script and will not be served by the static file handler
```

Resolution

This error can be caused by ASP.NET 4.5 not being correctly registered on the server. To correct this:

Windows Server 2012 or 2012 R2

Use the Server Manager to install ASP.NET 4.5.

1. Open the Server Manager.
2. Select **Manage > Add Roles and Features**. The Add Roles and Features wizard appears.
3. Click the **Next** button. The Select Installation Type page appears.
4. Click to select the **Role-based or feature-based installation for your server** selection button.
5. Click the **Next** button twice. The Select Server Roles page appears.
6. Click to select the **Web Server (IIS)** check box in the **Roles** list.
7. Click the **Next** button until you arrive at **Role Services** under **Web Server (IIS)**.
8. Drill down to **Web Server > Application Development** in the **Role Services** list.
9. Click to select the **ASP.NET 4.5** check box.
10. Click the Next button until you arrive at the final page.
11. Click the **Install** button.
12. Once installed, follow the resolution instructions in HTTP Error 404.2 - ISAPI and CGI Restrictions to ensure ASP.NET 4.0 is allowed to execute in IIS.

HTTP 404.2 Error ISAPI/CGI Restrictions Stopping .NET Framework 4.5.1

Error

An HTTP 404.2 error code is received when ISAPI/CGI Restrictions are preventing the.NET Framework 4.5.1 from running.

Resolution

1. Open Internet Information Services.
2. Select the Server in the left tree view.
3. In the **IIS** section, open ISAPI and CGI Restrictions.
4. For all items beginning with **ASP.NET v4.0**, right-click the item and select **Allow**.

Other Troubleshooting

This section encompasses a wide range of issues that don't fall under more specific categories. This section includes troubleshooting guides for various platforms and services, such as Google Authenticator setup, invalid Oracle data source length errors, and SAML configuration upgrade errors. It also covers issues like Linux connection errors, invalid domain errors, Quartz trigger job troubleshooting, VMware-related issues, and Windows local account access errors. Each folder offers targeted solutions to help users resolve uncommon or specialized technical problems efficiently.

Invalid Oracle Data Source Length

Error

Error Text

Oracle error: Invalid length for connection option 'Data Source', maximum length is 128.

Scenario

You run an SQL script that attempts to connect to an Oracle DB using the `System.Data.OracleClient.OracleConnectionStringBuilder.set_DataSource(String value)` call. The script runs from the local site as a database account, and an error occurs.

The error stack trace from the logs looks similar to this:

```
 2020-05-20 11:24:16,149 [CID:] [C:] [TID:PriorityScheduler Thread @ Normal] ERROR <Secret
Server URL />.PasswordChangers.DependencyChangers.Sql.SqlScriptRunner - Invalid length for
connection option 'Data Source', maximum length is 128.
System.ArgumentException: Invalid length for connection option 'Data Source', maximum
length is 128.
  at System.Data.OracleClient.OracleConnectionStringBuilder.set_DataSource(String value)
  at System.Data.OracleClient.OracleConnectionStringBuilder.set_Item(String keyword,
Object value)
  at System.Data.Common.DbConnectionStringBuilder.set_ConnectionString(String value)
```

```
  at <Secret Server URL
/>.PasswordChangers.DependencyChangers.Sql.SqlScriptRunner.BuildConnection
(DbProviderFactory dbProviderFactory, String connectionString, IDictionary2
additionalConnectionItems)
  at <Secret Server URL
/>.PasswordChangers.DependencyChangers.Sql.SqlScriptRunner.ExecuteScript(String
providerName, String connectionString, IDictionary2 additionalConnectionItems, String
commandText, IEnumerable`1 parameters)
```

Resolution

This error is an Oracle issue. Oracle has built-in string length restrictions on data source strings. One possible workaround is to use IP addresses for the Oracle server instead of FQDNs. Failing that, search the Internet for `"Invalid length for connection option 'Data Source', maximum length is 128"` to discover the latest information and workarounds.

Troubleshooting TOTP MFA for Secret Server Accounts

Google Authenticator generates tokens based on time synchronization. If Secret Server's clock is inaccurate or unsynchronized with Google Authenticator devices, user token validation may fail during enrollment or login.

Solution A (Preferred)

Ensure that the clock on Secret Server is accurate and synchronized with the device running Google Authenticator. Configure the web servers to synchronize their clocks with a reliable domain controller clock or an NTP server.

Solution B

By default, the token time leniency value is set to zero, meaning the token supplied must be completely accurate. Follow these steps to configure Secret Server to accept tokens that are slightly behind or ahead:

1. Open the `web-appSettings.config` file and add the following key between the `appSettings` tags:

   ```
   <add key="TOTPLeniency" value="0 or greater value here" />
   ```

2. Change the leniency value. We recommend setting this value to no higher than 2.
3. Recycle your IIS application pool. You must recycle your IIS application pool for the setting to take effect.

Troubleshooting Heartbeat and RPC Errors for Linux Secrets

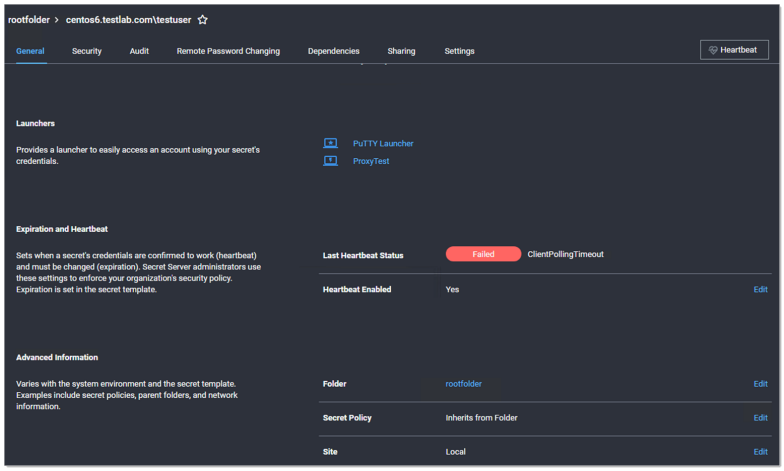
When using Secret Server for SSH password rotation, you may encounter errors when changing a secret. This article helps the reader troubleshoot the configuration of Remote Password Changing (RPC) in Secret Server to avoid errors.

See "Troubleshooting SSH Issues" on page 264 for other SSH issues.

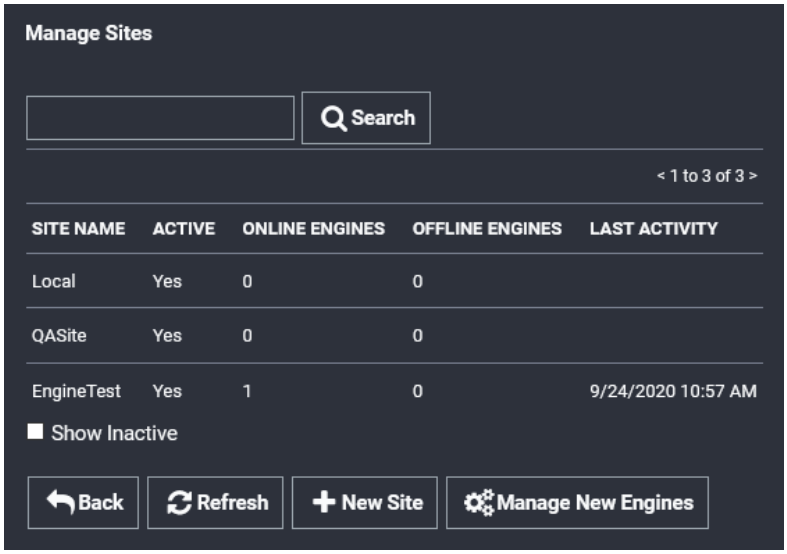
Step 1: Verifying Ports and Connectivity

To determine if the heartbeat issue is outside of Secret Server:

1. Open the secret which is failing Remote Password Changing in Secret Server.



2. Scroll down to the **Advanced Information** section. You may have to click the **Advanced** link.
3. Note the **Site** parameter.
4. Go to **Admin > Distributed Engine**.
5. Click the **Manage Sites** button. The Manage Sites page appears:



6. Click the **Site Name** link for the site. The Site View page appears.
7. Note the **Processing Location** parameter for the site.
8. If the processing location is **Local** and website processing is enabled, do your testing on the Secret Server application server. If it is **Distributed Engine**, do your testing on the distributed engine machine.
9. In PowerShell run one of the following command for the machine you are trying to connect to from the secret:

```
Test-NetConnection -ComputerName <computer_name> -Port 22
```

If you chose a custom port, note it–that port will need to be changed on the RPC too.

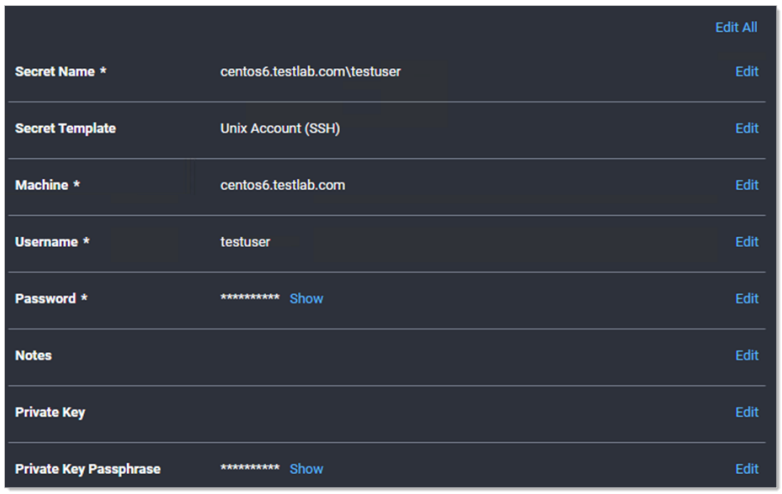
10. If the test was successful, proceed to the next step. If it was not successful, contact your networking team to open the port and test the connectivity. They can refer to "Ports and IP Addresses Used by Secret Server" on page 765.

Step 2: Testing Heartbeat and RPC in Secret Server

Procedure:

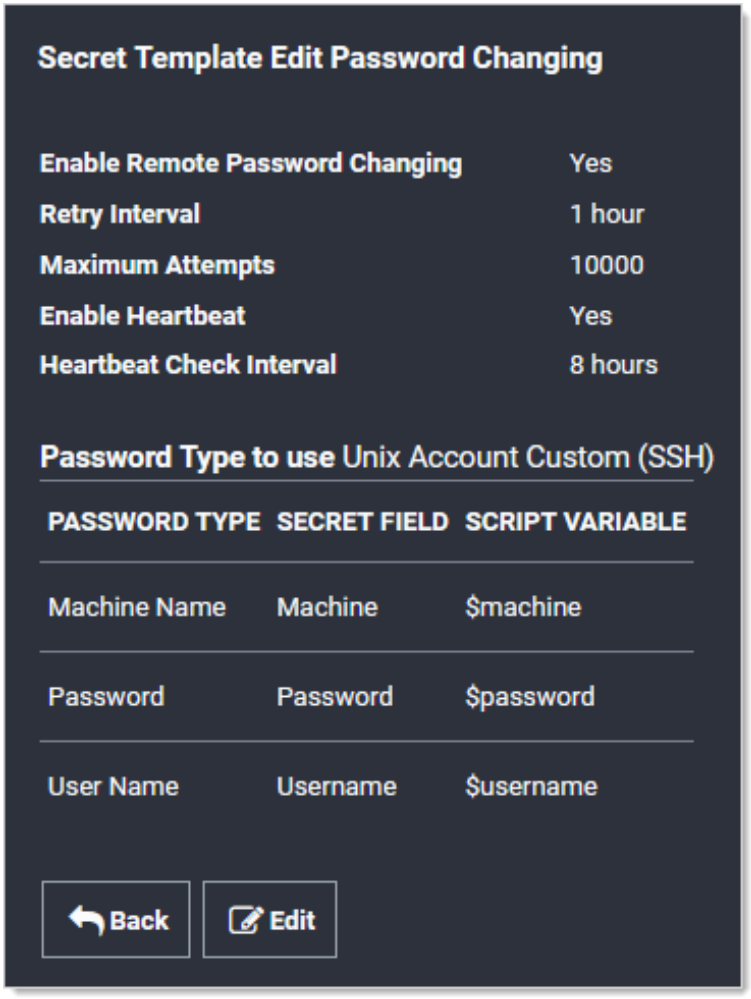
1. Return to the secret on Secret Server.
2. Click the **Remote Password Changing** tab of the secret (not shown).
3. Check the **Associated Secret** section to see if there is an associated account set on the secret for use with RPC:

4. Return to the **General** tab for the secret:



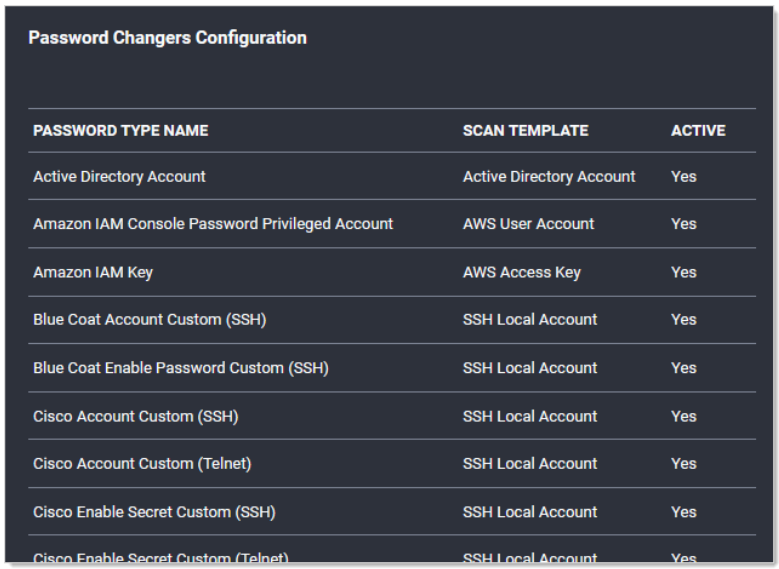
5. Note the **Secret Template** type.
6. Determine the password type for the template:
   a. Go to **Admin > Secret Templates**.
   b. Click to select the desired template in the dropdown list.
   c. Click the **Edit** button. The Secret Template Designer page appears (not shown).

d. Click the **Configure Password Changing** button at the bottom of the page. The Secret Template Edit Password Changing page appears:

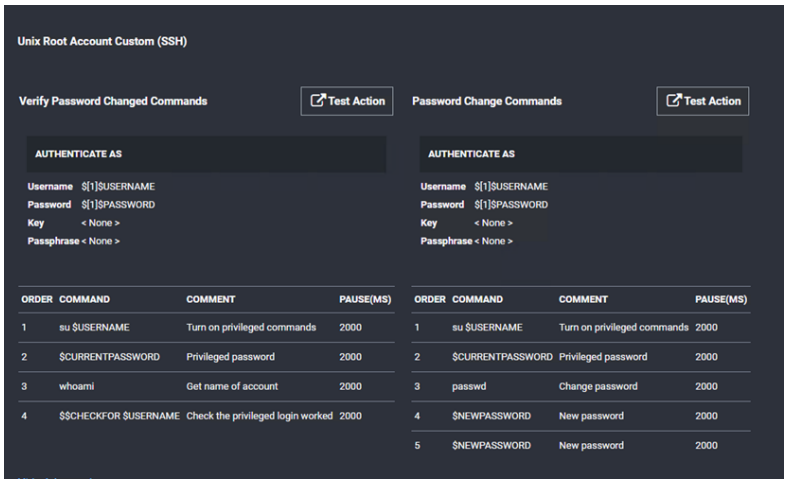


e. Note the password types used, the applicable secret field, and the equivalent script variable. These indicate reserved variables that reference fields in the secret, in this case, `$USERNAME`, `$MACHINE` and `$PASSWORD`. You will need to test your script using known-good values for these.

7. Go to **Admin > Remote Password Changing**. The Remote Password Changing Configuration page appears (not shown).
8. Click the **Configure Password Changers** button. The Password Changes Configuration page appears:

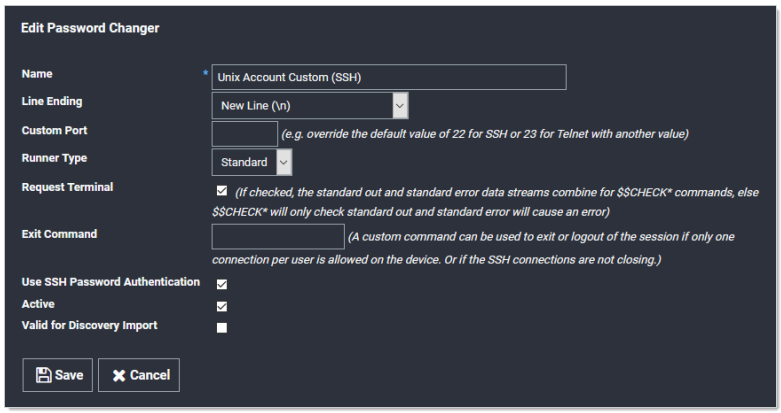


9. Click the name link for the same password changer. The password changer page for that changer appears:

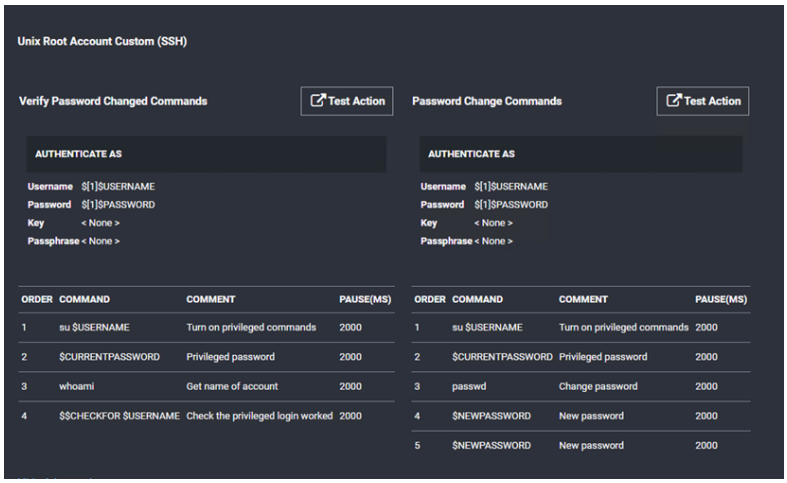


The **Verify Password Changed Commands** section defines the secret fields and commands to use to confirm that a password has rotated (changed) successfully on the target machine. The **Password Change Commands** section defines the secret fields and commands to use to change the password on the target machine.

10. Click the **Edit** button at the bottom of the page. The Edit Password Changer page appears:



11. If a port for the RPC is listed in the **Custom Port** text box, it must match the port that Secret Server connects to when running the commands seen on the previous page.
12. Click the **Cancel** button to return to the previous page:



13. The **Verify Password Changed Commands Test Action** button tests the defined password-changed verification listed under it. When clicked, it uses the "Authenticate As" parameters to connect to the accounts and run the commands to test for a heartbeat and check that the account and password is valid.

> This authenticates with a non-privileged associated secret and then uses that account to connect to the Linux machine. This is needed because root accounts are often unable to directly authenticate. Thus, several commands are run to test if the active account can be set to root. If that fails, heartbeat fails.

14. In the example command set for the section, when the heartbeat runs, the associated account ($[1]$USERNAME) authenticates, logs into the remote SSH device, and runs:
    a. `su $USERNAME` (The username from the secret)
    b. `$CURRENTPASSWORD` (The password which the password should have been changed to)

c. `whoami` (Returns the name of the active user, which indicates the su command and the provided parameters worked). This test checks that the returned username is the same as the username field in the secret. If it is not, the heartbeat fails.

d. `CHECKFOR $USERNAME` (Checks if the 'whoami' returns the username field from the secret. If it does not, an error is thrown and the heartbeat fails)

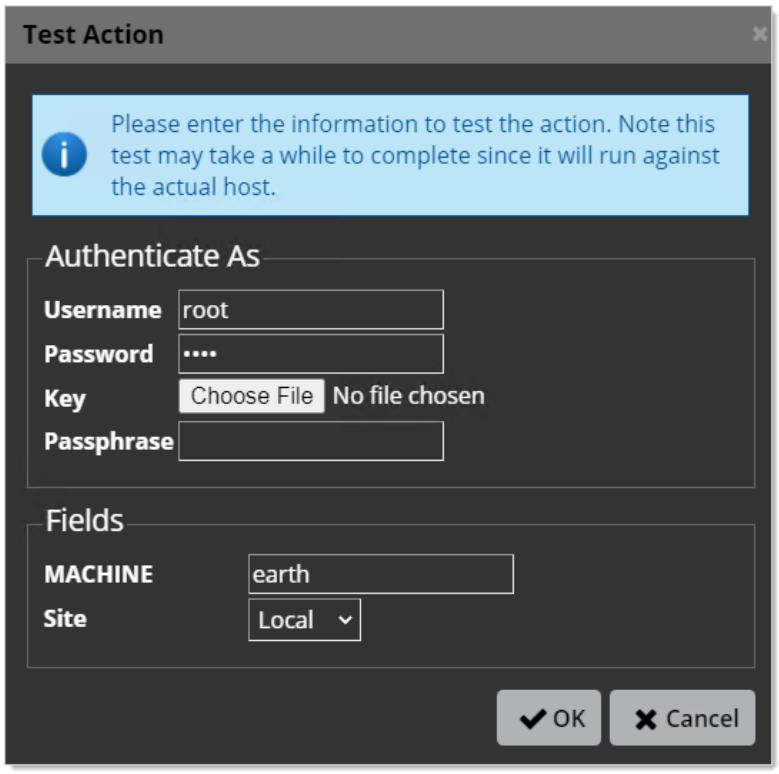
> Some of the command sets run by the "Verify Passwords Changed Test Action" button are empty. In that case, the test authenticates with the provided username and password, and if that is successful, so is the heartbeat. That is, the heartbeat uses the secret's own account (`$USERNAME`) and value to connect, rather than those of an associated secret.

> If the RPC is set up to use an associated secret but the secret does not have one, the secret fails to rotate and throws an error.

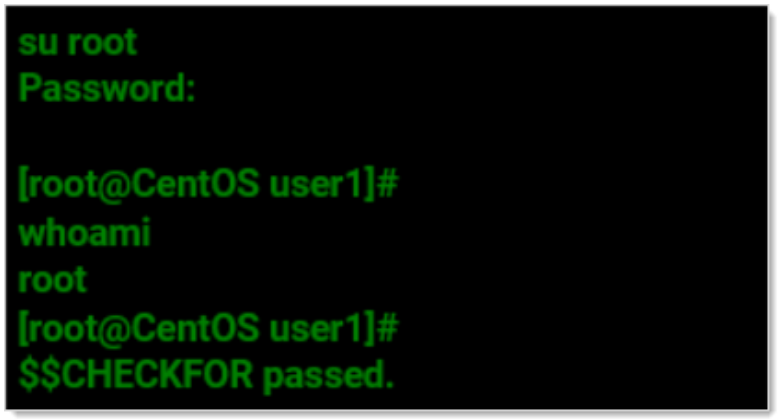
> For more on how Secret Server interprets what values to supply your custom script from the secrets involved, see "Editing Custom Commands" on page 1032 and "RPC Overview" on page 904.

> The heartbeat above is designed to authenticate with a non-privileged associated secret and use that account to connect to the Unix machine because root accounts are often unable to authenticate directly. Then, several commands are run to check if root can be successfully switched to. If this fails, the heartbeat fails.

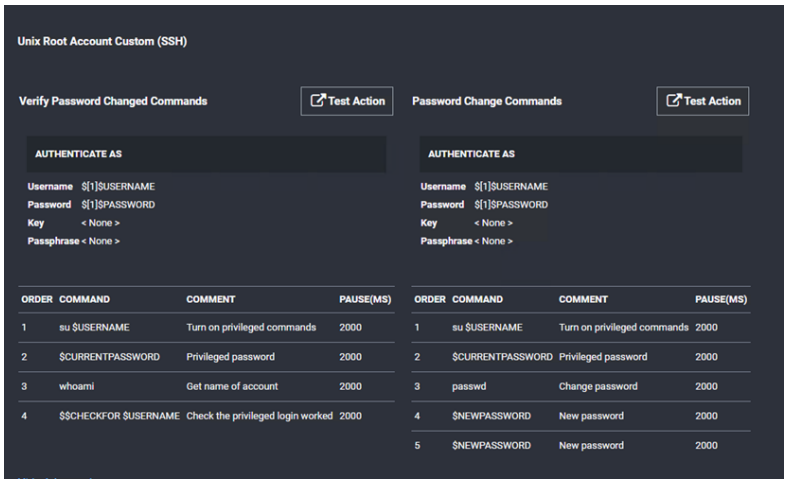
15. When you click the Verify Password Changed Commands **Test Action** button, the commands cannot read the fields from a secret or associated secret because when setting up the password changer no specific secret is calling it. Instead, for the test only, you manually provide the input parameters from the secret and associated secrets involved with the RPC. For example, the `$USERNAME` field refers to the user on the secret that you are trying to change. Whereas `[1][1]USERNAME` refers to the first associated secret linked to that secret.
16. When you click the button a popup appears for you to do just that:



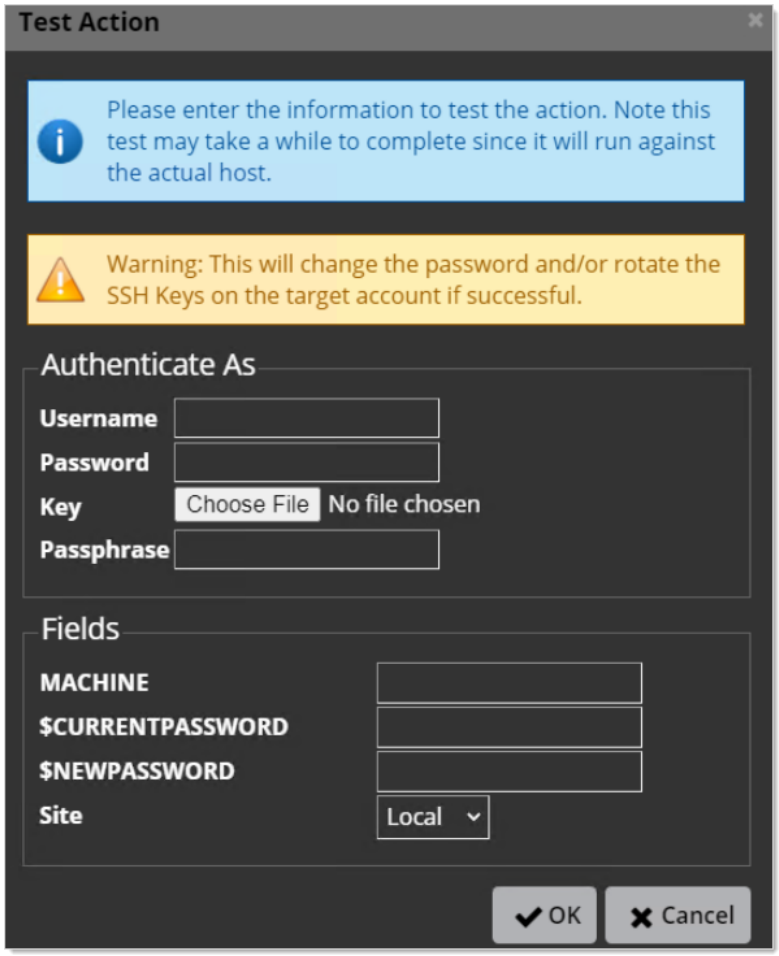
17. Type or select your parameters.
18. Click the **OK** button. The password-changed command set is tested with a simulated heartbeat, using what you entered. If any errors occur, record them for troubleshooting later. The console outputs something similar to this:



You then return to the previous page:



19. The **Password Change Commands** section defines the secret fields and commands to use to rotate (change) a password on the target machine. We now run a similar test on it.
20. Click the Password Change Commands **Test Action** button. Another Test Action popup page appears:



> Clicking the OK button in the following instructions **really changes the password or rotates the SSH keys on the target account** (the `$NEWPASSWORD` parameter gets changed), so record what you change it to, and update the secret with the new password (assuming the RPC is successful).

21. Similar to the last test, manually provide the input parameters. See Step 2: Testing Heartbeat and RPC in Secret Server for a description of how to fill in the parameters.
22. Click the **OK** button. The test connects with the "Authenticate As" accounts and runs the commands to change the password. A password rotation occurs, and more console output appears. Record any errors and output.
23. If the rotation did not occur, check the information that was presented to the changer from your secret. It is possible that the secret's data is involved in the issue.

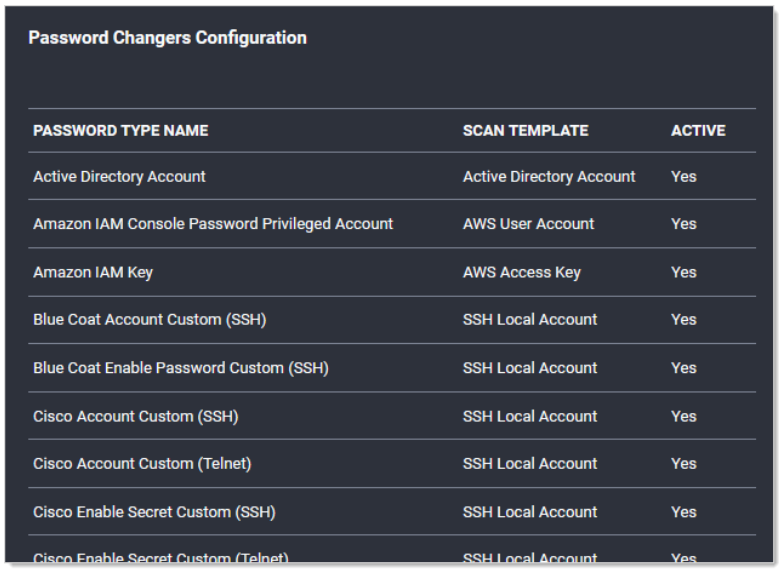
Step 3: Troubleshooting Heartbeat or RPC Outside of Secret Server

This section troubleshoots the commands used by Secret Server to heartbeat and RPC outside of Secret Server. The intent is to confirm a successful authentication and password change on the endpoint when the same commands are issued outside of Secret Server. If they do not, the commands must be revised to work within Secret Server.

1. Use the procedure from Step 1 to determine which machine (Secret Server application or DE) to perform this step on.
2. If you did not already, Download PuTTY on the application or any of the DE servers.
3. Open a browser tab on the secret which is failing to Heartbeat or RPC.
4. Do the same for each associated secret of that secret.

> Instead of the following three steps, you can instead go to ...\SecretServer\CustomCommandsEdit.aspx.

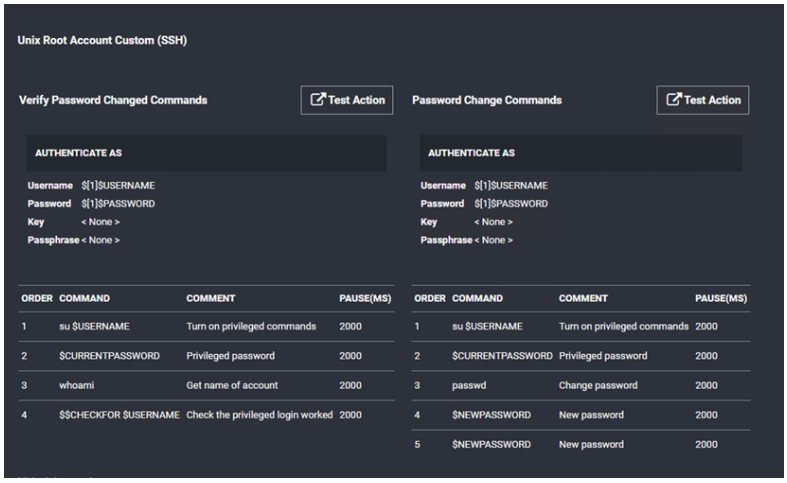
5. Go to **Admin > Remote Password Changing**. The Remote Password Changing Configuration page appears (not shown).
6. Click the **Configure Password Changers** button. The Password Changers Configuration page appears:



Password Changers Configuration

PASSWORD TYPE NAME	SCAN TEMPLATE	ACTIVE
Active Directory Account	Active Directory Account	Yes
Amazon IAM Console Password Privileged Account	AWS User Account	Yes
Amazon IAM Key	AWS Access Key	Yes
Blue Coat Account Custom (SSH)	SSH Local Account	Yes
Blue Coat Enable Password Custom (SSH)	SSH Local Account	Yes
Cisco Account Custom (SSH)	SSH Local Account	Yes
Cisco Account Custom (Telnet)	SSH Local Account	Yes
Cisco Enable Secret Custom (SSH)	SSH Local Account	Yes
Cisco Enable Secret Custom (Telnet)	SSH Local Account	Yes

7. Click the name link for the subject password changer. The password changer configuration page for that changer appears:



Unix Root Account Custom (SSH)

Verify Password Changed Commands Test Action

AUTHENTICATE AS

Username $[1]$USERNAME
Password $[1]$PASSWORD
Key < None >
Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	su $USERNAME	Turn on privileged commands	2000
2	$CURRENTPASSWORD	Privileged password	2000
3	whoami	Get name of account	2000
4	$$CHECKFOR $USERNAME	Check the privileged login worked	2000

Password Change Commands Test Action

AUTHENTICATE AS

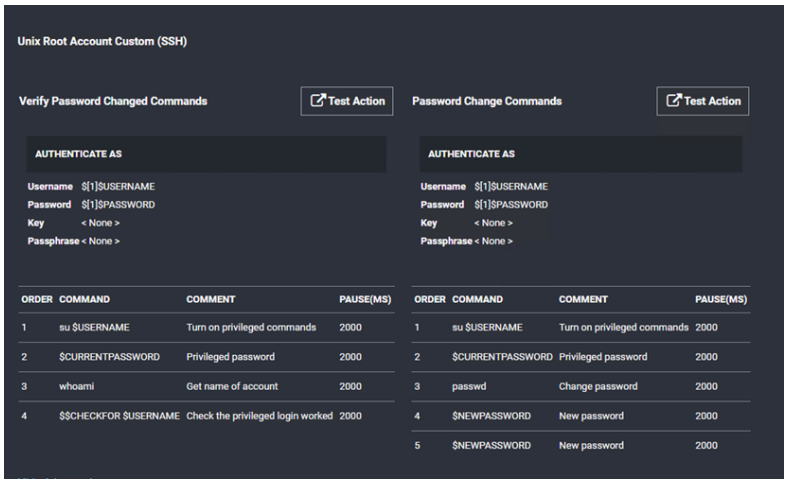
Username $[1]$USERNAME
Password $[1]$PASSWORD
Key < None >
Passphrase < None >

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	su $USERNAME	Turn on privileged commands	2000
2	$CURRENTPASSWORD	Privileged password	2000
3	passwd	Change password	2000
4	$NEWPASSWORD	New password	2000
5	$NEWPASSWORD	New password	2000

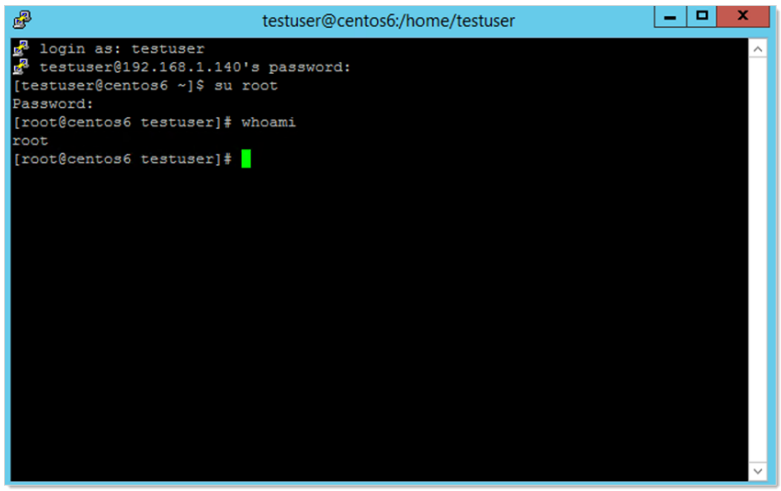
8. Determine if you are troubleshooting heartbeat or RPC: The **Verify Password Change Commands** section applies to heartbeat, and the **Password Change Commands** section applies to RPC. Which you use (or both) depends on what failed in Step
9. Return the Secret Server or DE server you are testing.
10. Launch PuTTY.
11. Type the host name or IP address of the subject Linux machine (generally, the Machine field in the secret).
12. Log on with the username and password for the main or associated secret.

> If you are successful with connecting with PuTTY but not Secret Server, launch PuTTY in debug mode and collect a log file. Determine what cipher was used to connect. If you have a machine that works with Secret Server, compare the ciphers. Also check if the endpoint handles interactive logins differently. Secret Server's logins for RPC are non-interactive. See "Troubleshooting SSH Issues" on page 264 for more information about troubleshooting connection issues in Putty.

13. Use the commands listed in the "Authenticate As" section you are troubleshooting directly in PuTTY to determine if they work outside of Secret Server. For example, given these heartbeat (Verify Password Changed) commands:



Your console would look like this:



In this example, we assumed the secret contained a value of "root" for the Username field, and the associated account in the first position was "testuser." This example was successful because the `$$CHECKFOR $USERNAME` found "root" on the previous line.

If the `su root` command were to fail above and reports the message "Username is not in the sudo users file. This incident will be reported." then the `$$CHECKFOR` would fail and the heartbeat would fail to verify. This type of issue needs to be remediated on the endpoint.

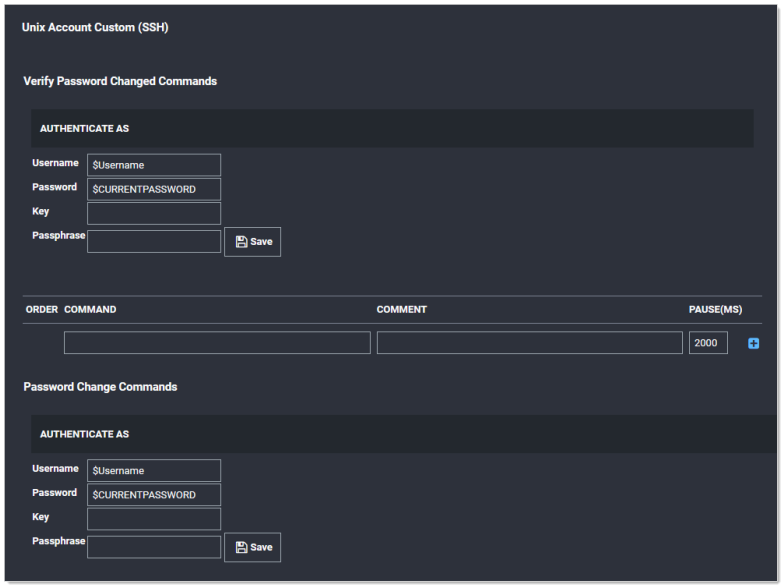
14. If the issue is clearly an endpoint issue, remediate it and repeat the commands in PuTTY.
15. Once the commands work properly in PuTTY, if the RPC or heartbeat command set needs adjustment to match the working PuTTY equivalent, return to Secret Server and make the changes to the command set (see the next step).

> Before you change the RPC commands, ensure that the device that you are working on belongs to the secret template you are using. Secret templates dynamically update all the secrets based on them, so **all secrets with this template are affected by your changes**. We strongly recommend that if this device is unique or you are storing an independent root account in the associated secret template, you should:

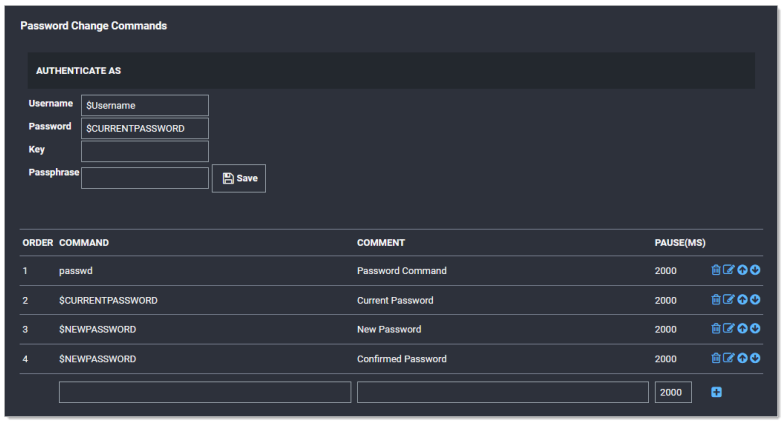
   a. Copy the template you are using, giving the copy a descriptive name.
   b. Create a new password changer based on the current one that you are using.
   c. Assign it the secret template you just created.

   This ensures that you do not change how ALL devices related to a secret template when you only intend to change a single devices. Accounts that are the same type on the same device should share the same template.

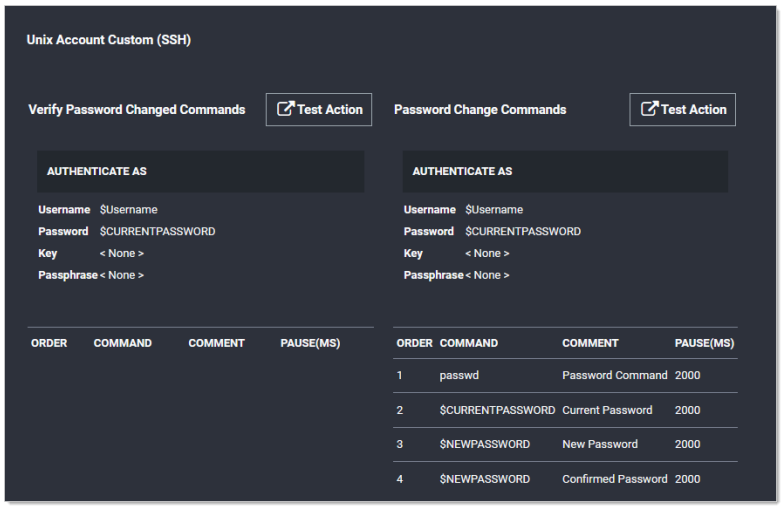
16. Click the **Edit Commands** button at the bottom of the subject password changer page. The commands for RPC and heartbeat appear:



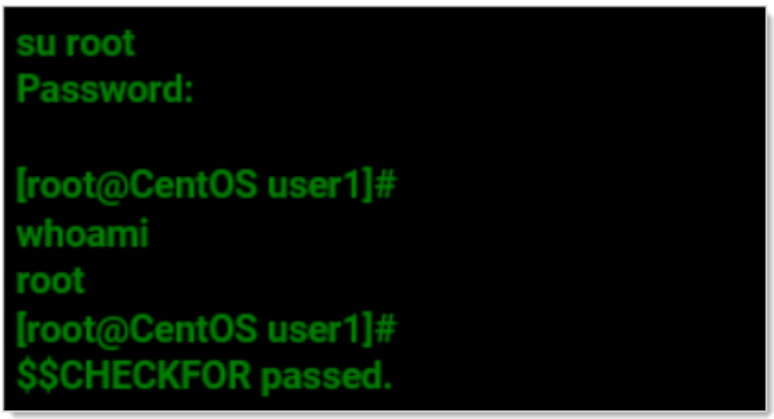
17. Scroll down to the command list in the **Password Change Commands** section:



18. Click the blue edit icon to the right of any commands you want to change. The command becomes editable.
19. Edit the command to make it match your known-good revision.
20. Click the blue save icon next to the amended command.
21. Click the **Back** button to return to the password changer page:



22. One of the Test Action popups appears.
23. Type the known-good values from the secret in the text boxes.
24. Click the **OK** button to test heartbeat or RPC. The result should look something like this:



Troubleshooting Invalid Domain Errors

This topic discusses resolving the "The specified domain is not a valid domain" error.

Troubleshooting Procedure

1. Verify that you are entering the fully qualified domain name in the domain field and that the domain username and password fields are correct.
2. Ensure that the ports used for LDAP (389) or LDAPS (389 and 636) are open. For more information about the ports used by Secret Server, see "Ports and IP Addresses Used by Secret Server" on page 765.
3. Ensure that your server is connecting to the correct DNS server:
   a. Open the command console as an administrator (**Start > Run > cmd**).
   b. Type `ipconfig /all`.
   c. Press **<Enter>**.
   d. Find your primary Ethernet adapter and look in the **DNS Servers** section. Verify that the DNS server is correct.
4. If the DNS server is incorrect, then follow these steps to configure the DNS server:
   a. Open up your control panel (**Start > Control Panel**).
   b. Click on **Network and Sharing Center**.
   c. Click **Manage Network Connections** on the left.
   d. Right click on your primary network adapter and select **Properties**.
   e. Click **Internet Protocol Version 4 (TCP/IPv4)**.
   f. Click **Properties**.
   g. Click to select the **Use the following DNS server addresses selection** button.
   h. Type your primary DNS server in the first row.

i. If you have a secondary DNS server, put it in the second row.

> Both DNS servers must contain the SRV record for your domain controller.

5. Check that your server is retrieving domain controller DC records correctly:
   a. Open up your control panel (**Start > Control Panel**).
   b. Type `nslookup`.
   c. Press **<Enter>**.
   d. Type `set q=srv`
   e. Press **<Enter>**.
   f. Type `_ldap._tcp.dc._msdcs.<Fully_Qualified_Active_Directory_Domain_Name>`.
   g. Press **<Enter>**.
   h. If you get a result that looks like:

      ```
      _ldap._tcp.dc._msdcs.<Fully_Qualified_Active_Directory_Domain_Name> SRV service
      location: priority = 0 weight = 100 port = 389 svr hostname = *Domain_Controller_Host_
      Name*
      ```

      Then you are retrieving the DNS record correctly. Otherwise, your DNS records are not correctly configured.

Configuring the DNS Record on Your Server

1. If you are **not** using a Windows DNS server, contact your vendor to ask how to add SRV records. You will need to add a SRV record pointing `_ldap._tcp.dc._msdcs.<Fully_Qualified_Active_Directory_Domain_Name>` to your primary DNS server.
2. Connect to your Windows DNS server.
3. Open the DNS control panel (**Start >Administrative Tools > DNS**).
4. Expand the node corresponding to your server.
5. Expand the **Forward Lookup Zones** node.
6. Expand the node corresponding to your domain.
7. Delete the **_msdcs** node if it exists.
8. Right click on the domain node and select **New Domain...**
9. Type`_msdcs` as the name.
10. Right click on the new **_msdcs** node, and select **New Domain...**
11. Type `dc` as the name.
12. Right click on the new **dc** node and select **Other New Records...**
13. Select **Service Location (SRV)** as the record type.
14. Click the **Create Record** button.
15. Select **_ldap** as the service.

16. Select **_tcp** as the protocol.
17. Type `389` as the port.
18. Type the fully qualified host name of your DC or the IP address in the **Host offering this service:** text box.
19. Click the **OK** button.
20. Click the **Done** button.
21. Open up the services console (**Start > Run > services.msc**)
22. Right click on the **DNS Server** service and select **Restart**. Your domain DNS record should now be set up.

Resolving Other DNS Issues

Secret Server requires that the DNS is correctly configured to add a domain. For additional tips on tracking down DNS Issues, see this Troubleshooting Active Directory Installation Wizard Problems.

Also ensure the domain controller is using the appropriate DNS. The `ipconfig /registerdns` command (as per the link above) is frequently helpful for entering the correct DNS entries in for a given domain.

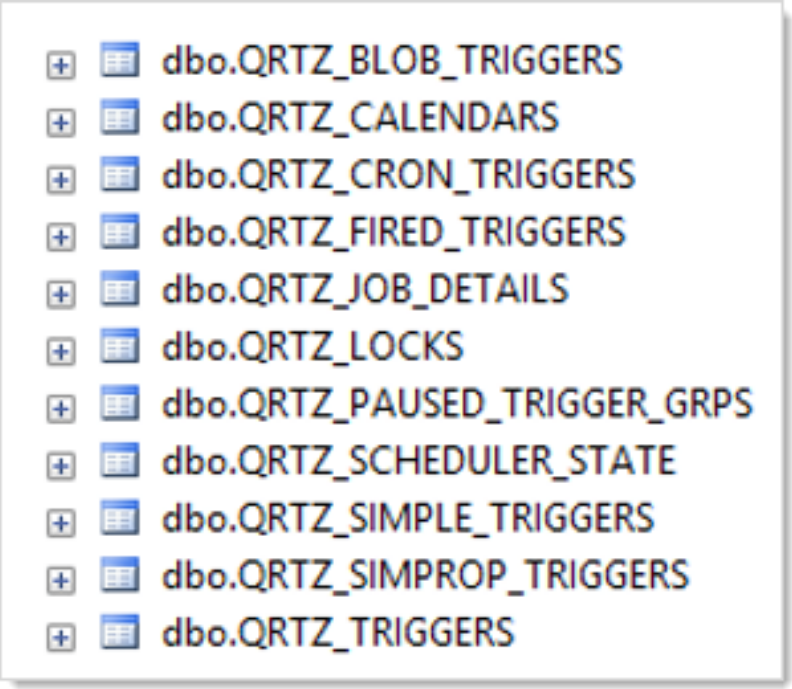
Troubleshooting Quartz Trigger Jobs

This topic only applies to **Secret Server On-Premises**.

In Secret Server 10.7 and newer, all background operations have moved to Quartz. All scheduled background operations have become jobs registered in tables inside the Secret Server database.

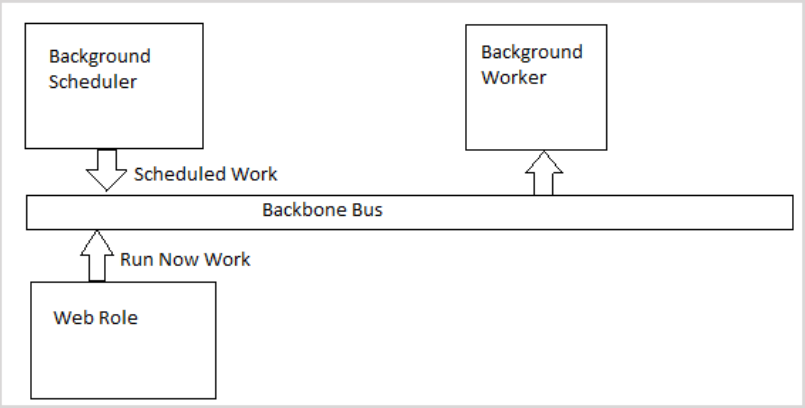
Quartz-Related Tables

All Quartz-related table names start with "QRTZ_ "



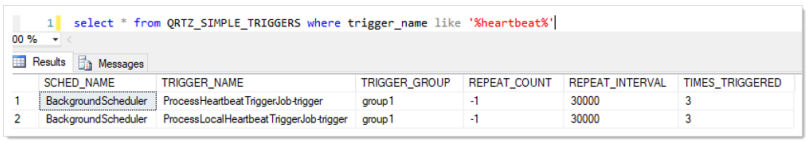
Trigger Job Components

Each background job runs as a consumer, similar to how a distributed engine (DE) works, in the background worker. Each job has a corresponding trigger job registered in Quartz. Quartz executes the trigger job at the configured time and interval. The trigger job puts a message on the backbone bus, which is picked up and executed by the background worker:



Simple Trigger Job

There are two kinds of registered trigger jobs, *simple triggers* and *cron triggers*. A simple trigger fires on an interval. For example, on premises the heartbeat job runs every 30 seconds:



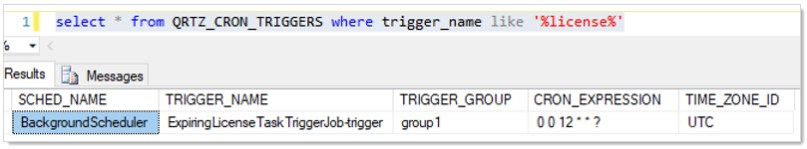
```
select * from QRTZ_SIMPLE_TRIGGERS where trigger_name like '%heartbeat%'
```

	SCHED_NAME	TRIGGER_NAME	TRIGGER_GROUP	REPEAT_COUNT	REPEAT_INTERVAL	TIMES_TRIGGERED
1	BackgroundScheduler	ProcessHeartbeatTriggerJob-trigger	group1	-1	30000	3
2	BackgroundScheduler	ProcessLocalHeartbeatTriggerJob-trigger	group1	-1	30000	3

In this case, you can ignore the TRIGGER_GROUP and REPEAT_COUNT columns. The latter column limits the number repetitions, which we do not do. The REPEAT_INTERVAL is in milliseconds, so 30000 is 30 seconds. TIMES_TRIGGERED is a counter of trigger firings. There are two trigger jobs for heartbeat. The local job handles secrets that do not run through a distributed engine while the other heartbeat handles secrets that do run through a distributed engine.

Cron Trigger Job

Cron jobs run at a specific time using a variation of the well-known cron syntax. For example:



```
select * from QRTZ_CRON_TRIGGERS where trigger_name like '%license%'
```

SCHED_NAME	TRIGGER_NAME	TRIGGER_GROUP	CRON_EXPRESSION	TIME_ZONE_ID
BackgroundScheduler	ExpiringLicenseTaskTriggerJob-trigger	group1	0 0 12 * * ?	UTC

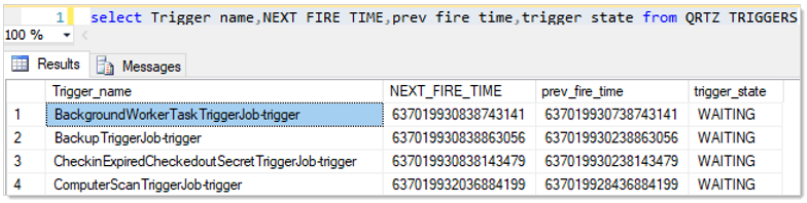
In this example, the trigger ExpiringLicenseTaskTriggerJob runs every day at 12:00 PM UTC.

The Quartz jobs are run within the background scheduler role, which runs on every node that has the background-worker-enabled bit set to true. Quartz coordinates trigger-job execution through the database to ensure that the same trigger-job is run just once, even if the Quartz scheduler is running on multiple nodes simultaneously.

Procedures

Viewing Trigger Job State (QRTZ_TRIGGERS Table)

The current state of each trigger-job can be seen in the QRTZ_TRIGGERS table:



```
select Trigger name,NEXT FIRE TIME,prev fire time,trigger state from QRTZ TRIGGERS
```

	Trigger_name	NEXT_FIRE_TIME	prev_fire_time	trigger_state
1	BackgroundWorkerTaskTriggerJob-trigger	637019930838743141	637019930738743141	WAITING
2	BackupTriggerJob-trigger	637019930838863056	637019930238863056	WAITING
3	CheckinExpiredCheckedoutSecretTriggerJob-trigger	637019930838143479	637019930238143479	WAITING
4	ComputerScanTriggerJob-trigger	637019932036884199	637019928436884199	WAITING

Viewing Trigger Firing Times

In the previous figure, the NEXT_FIRE_TIME and PREV_FIRE_TIME columns are not human readable. To convert them to something you can read, run the following SQL query:

```
SELECT TOP 1000
    [TRIGGER_NAME]
    ,[TRIGGER_STATE]
    ,[TRIGGER_TYPE]
```

```
    ,NEXT_FIRE_IN = DATEDIFF(second, GETUTCDATE(), CAST(NEXT_FIRE_TIME/864000000000.0 -
693595.0 AS DATETIME))
    ,UTCNOW  = GETUTCDATE()
    ,NEXT_FIRE_TIME2 = CAST(NEXT_FIRE_TIME/864000000000.0 - 693595.0 AS DATETIME)
    ,[PREV_FIRE_TIME2] = CAST(PREV_FIRE_TIME/864000000000.0 - 693595.0 AS DATETIME)
    ,[START_TIME2] = CAST(START_TIME/864000000000.0 - 693595.0 AS DATETIME)
    ,[END_TIME2] = CAST(END_TIME/864000000000.0 - 693595.0 AS DATETIME)
    ,[PRIORITY]
FROM [dbo].[QRTZ_TRIGGERS]
```

This query shows the NEXT_FIRE_TIME in seconds and the other fields as DateTimes, along with the current UTC date to make it easier to compare to the current time.

The TRIGGER_STATE is normally WAITING. If the trigger job is currently running, the state will be ACQUIRED. If there was an error running the trigger job, the state will be ERROR. Once a trigger job enters an ERROR state it will not run again. To address this, Secret Server automatically looks for any trigger jobs in the ERROR state every 10 minutes and changes their state to WAITING. If the NEXT_FIRE_TIME is a large negative number, it indicates that the scheduler role is not running–take a look at the Secret Server-BSSR.log file to see its status.

Adjusting Trigger Job Frequency

Secret Server populates the Quartz schedules when it starts up and creates them if they do not exist. Secret Server does not update pre-existing schedules, so it is possible to change the frequency of a job in the table and have the system run with that frequency from that point on.

Do not change the job frequency unless it is absolutely necessary.

Delinea controls those schedules and in future releases may not recognize changes you have made to those schedules. Furthermore, we do not test with a variety of schedules and have no plans to do so. Therefore, adjusting the schedules is a risky undertaking that may cause issues within Secret Server. If you must change the tables, you must bounce IIS on all the nodes where the scheduler runs, after you change the table values.

To trigger an infrequent task just once, without altering its schedule, follow these steps:

1. Update the NEXT_RUN_TIME value in the QRTZ_TRIGGERS table to match the NEXT_RUN_TIME value from a trigger that fires frequently, such as one of the heartbeat triggers.
2. Recycle the scheduler role by running Recycle Background Processes on the Diagnostics page,
3. Reset the background worker (inside the application pool) in one of the following ways:
   - Click Recycle Background Processes on the Diagnostics page. This shuts down and restarts the scheduler.
   - Recycle the application pool. This restarts Secret Server in its entirety.
   - Reset IIS. From Secret Server's perspective this is nearly the same as the second option.

The PDF version of this online document is automatically generated and thus may have minor formatting anomalies.

This document is not updated with every release–many releases do not affect the guide's contents and thus do not warrant a document update.

Troubleshooting SAML Configuration Errors After Upgrading

This topic is for upgrades of Secret Server from a version earlier than 10.2.

Initial Troubleshooting

Changes to the `saml.config` were introduced in Secret Server 10.2. Secret Server should automatically convert the existing `saml.config` to the latest format. If it does not:

1. Ensure that the application pool has write access to the `saml.config` file.
2. Restart the application pool in IIS and try to log in again.
3. If Secret Server is running in a clustered environment:
   a. Copy the `saml.config` from the Web node that was upgraded to the remaining web nodes.
   b. Restart their application pools in IIS.

If that does not resolve the issue or Secret Server is not running in a clustered environment, there may be some other issue that prevented the `saml.config` from converting successfully during the upgrade. Please contact "Technical Support" on page 72 for assistance.

See the "Configuring SAML Single Sign-on" on page 422 article for more information on configuring your saml.config in 10.2

Additional Troubleshooting

If the `saml.config` is not loading properly, there are a few possibilities:

- The `saml.config` file is invalid. Ensure that it contains valid XML. Element and attribute names are case sensitive. Ensure that the elements and attributes names and value are valid for SAML configuration.

  See the `saml.config.template` file in Secret Server's root folder for guidance on which elements and attributes can be used.

- Secret Server is running in a clustered environment and some nodes are not yet configured. Copy the `saml.config` from the functioning Web node to all of the remaining Web nodes and restart their Application Pools in IIS.

Restart the Application Pool in IIS any time changes are applied to the `saml.config` file. If issues remain after following these steps, please contact "Technical Support" on page 72 for assistance.

Troubleshooting SSH Issues

When troubleshooting performance or connectivity issues with SSH with or without proxy it is useful to enable SSH debug logging on your remote host. There are several things that could go wrong during the connection process

and the SSH debug log tells you how far the connection gets before failing. To enable debug logging on a host the SSH service should be started with the debug flag.

> A UNIX administrator should be tasked with these operations because if the box is a remote host with no local access then an incorrect action could leave you locked out of the machine if SSH is the only remote connection possible.

Local Servers with Direct Access

The following example works best with a local connection. You can start the SSH service in verbose debug mode where the debug output is sent to a file on the local system and the service terminates after the remote connection ends with the following:

```
/usr/sbin/sshd -ddd 2> sshd-debug.log
```

Remote Servers

Another way to configure SSH to log debugging information is to have syslog set up. You will need to add a syslog entry for the SSH service in `/etc/syslog.conf`:

```
*.* /var/log/sshd/sshd.log
```

And then configure the SSH service to have a log level of DEBUG3, which can be modified in `/etc/ssh/sshd_conf`:

```
LogLevel DEBUG3
```

Then restart the SSH service:

```
service sshd restart
```

> On some systems, the log may already be configured to output to `/var/log/auth.log`.

Logging from the Client Perspective

You can also do logging from the perspective of the client connection to the remote host. This sort of logging helps you to understand what a normal successful connection should look like. To obtain logging from the client connection, you can run SSH in verbose mode.

```
ssh -vvv user@host
```

The debug information will be sent to the console. Or if you are using PuTTY, then you can right click the PuTTY window title after connecting to the remote host and selecting "Event Log".

Understanding SSH Logging

Example

The following is an example of standard debug output from PuTTY looks like the following:

1. The client begins by looking up the hostname and see if the host is valid:

   ```
   2016-01-07 12:23:57 Looking up host "192.168.1.60"
   ```

2. The client proceeds to make a TCP connection to the host:

```
2016-01-07 12:23:57 Connecting to 192.168.1.60 port 22
```

3. The client sends a message to the server saying what version of SSH we are using. In this example we are using SSH 2 over PuTTY v0.65:

   ```
   2016-01-07 12:23:57 We claim version: SSH-2.0-PuTTY_Release_0.65
   ```

4. The server sends back a message saying what version of SSH they are using. In this example we are connecting to an Ubuntu Server running an SSH 2 over OpenSSH v6.6.1p1:

   ```
   2016-01-07 12:23:57 Server version: SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2
   2016-01-07 12:23:57 We believe remote version has SSH-2 channel request bug
   ```

5. The client confirms the type of connection that can be used with the server. In this example the communications will be done using SSH 2:

   ```
   2016-01-07 12:23:57 Using SSH protocol version 2
   ```

6. The client begins the key exchange process and agrees on using Diffie-Hellman Group Exchange 256:

   ```
   2016-01-07 12:23:57 Doing Diffie-Hellman group exchange
   2016-01-07 12:23:57 Doing Diffie-Hellman key exchange with hash SHA-256
   ```

7. The server replies with their host key fingerprint information to identify its identity and aide in the prevention of Man in the Middle attacks. In this example the server is presenting an RSA 2048-bit key:

   ```
   2016-01-07 12:23:57 Host key fingerprint is:
   2016-01-07 12:26:53 ssh-rsa 2048 e0:d4:94:36:e9:20:fd:e3:58:ad:8d:4c:4a:1f:27:e8
   ```

8. The client initializes the transport layer encryption using AES-256 with SDCTR enabled and uses SHA-256 for message authentication:

   ```
   2016-01-07 12:26:53 Initialized AES-256 SDCTR client->server encryption
   2016-01-07 12:26:53 Initialized HMAC-SHA-256 client->server MAC algorithm
   ```

9. The server initializes the transport layer encryption using AES-256 with SDCTR enabled and uses SHA-256 for message authentication:

   ```
   2016-01-07 12:26:53 Initialized AES-256 SDCTR server->client encryption
   2016-01-07 12:26:53 Initialized HMAC-SHA-256 server->client MAC algorithm
   ```

10. The client sends a password.

    ```
    2016-01-07 12:27:12 Sent password
    ```

11. The server validates the password and granted access to the user:

    ```
    2016-01-07 12:27:12 Access granted
    ```

12. The session opens a shell for user interaction:

    ```
    2016-01-07 12:27:12 Opening session as main channel
    2016-01-07 12:27:12 Opened main channel
    2016-01-07 12:27:12 Allocated pty (ospeed 38400bps, ispeed 38400bps)
    2016-01-07 12:27:12 Started a shell/command
    ```

Confirming Proper Operation

Things to look for in an SSH log:

- Verifies the host IP address that you are connecting to.
- Verifies the port is correct for the address that you are connecting to.
- Verifies that you are not using an outdated SSH client.
- Verifies the SSH protocol you are using.
- Verifies what group exchange algorithm is being used.
- Verifies the server identity using the presented fingerprint. If the fingerprint is not expected then there may be malicious server between you and the remote host you want to connect to. Alert the administrator to verify if the host key has changed or if there is another issue.
- Verifies the transport layer and HMAC ciphers being used.
- Verifies that the password or key being sent is accepted by the server.
- If a connection does not open, it notes what the last successful step was and then what the next failed step is to find what the issue is.
- If using Secret Server proxy, it is useful to collect the client-to-proxy SSH log and then the proxy-to-remote-host log from the remote server.

VMware Issues

Secret Server supports operating systems in a VMware virtual machine environment in an identical manner as it runs on any other major x86-based systems without initially requiring reproduction of issues on native hardware. Should Delinea Support suspect that the virtualization layer is the root cause of an incident, you must contact VMware support provider to resolve the VMware issue. While Delinea products are expected to function properly in a VMware virtual environment, there may be performance implications that can invalidate Secret Server sizing and recommendations.


Migrating between CPU families or versions will require re-activation.

Windows Local-Account Access-Denied Error Workaround PowerShell Scripts

Overview

Beginning with Windows 10 version 1607 (Creator's Update) and Windows Server 2016, the default GPO security descriptor denies users remote access to Security Account Manager (SAM) with non-domain credentials, and therefore prevents remote heartbeat and password changes made by otherwise-authenticated local user accounts. Affected Windows local account secrets return "Access Denied" on a heartbeat or remote password change.

This article provides a script and instructions to address these "access denied" errors. The script modifies the default local group policy remote SAM access security descriptor to allow all local users on a specified machine remote SAM access after authentication. This script requires elevated PowerShell permissions.

Adding an account to the local computer's Administrators group does not solve the problem.

On most systems, the Administrators group on the local machine is part of the "Network Access: Restrict clients allowed to make remote calls to SAM" security policy setting. Through testing, we determined that Windows currently treats this group as only the built-in administrator account for this configuration. Therefore, if you add another user to the Administrators group on the machine, that user will be unable to heartbeat since it is not the

built-in administrator account. In addition, the built-in object, "Local account and a member of Administrators" does not allow a local account that is a member of Administrators to heartbeat for any account other than the built-in administrator account.

Additional Requirements

For heartbeat to work correctly, make sure that the local or authenticated users are:

- *Not* in the "Deny access to this computer from the network" security policy
- *In* the "Access this computer from the network" security policy

For Built-In Administrator accounts, also disable the user account control by setting the User Account Control to "Use Admin Approval Mode for the built-in Administrator account."

Remediation Options

Option 1: Creating a custom group and adding users to it (this is what the script does for users on the endpoint) then adding that group to the security setting to allow the user to heartbeat successfully. New local users need to be added to the custom group if they are created in the future.

Option 2: Adding a user individually to the security setting to allow the user to heartbeat successfully.

Option 3: Modifying the Default GPO: Adding "allow authenticated or local users" to the security setting. This allows all local users or all users who are authenticated to the machine to bypass this setting. This does requires the PowerShell Script below. The drawback is that this allows all users to remotely access SAM, so long as they are authenticated.

Option 4: Create a heartbeat workaround for GPO "Network Access: Restrict Clients Allowed to Make Remote Calls to SAM." This is addressed in the last section. This is for situations where the GPO needs to be completely bypassed.

Option 3: Modifying the Default GPO

PowerShell Script Description

This script adds a local non-privileged user group to the machine (a custom group name can be specified with the `-GroupName` parameter), adds all local users to the group, and then adds this group to the "Network Access: Restrict clients allowed to make remote calls to SAM" local group policy. This allows all local users within the group remote access to SAM after authentication, which is required for Secret Server heartbeat and password changing.

Download

Extract the .ps1 script found here: https://updates.thycotic.net/secretserver/support/PowerShell_Win10-HB-RPC-Fix/Win10-HbFix.zip. Run in an elevated PowerShell ISE session.

Script Argument Help

Command Prompt Help

For full help text, run:

```
> Get-Help C:\Script\Win10-HbFix.ps1 -Examples
```

Parameters

-ComputerNames (string[])

Specifies the computers on which the script runs (comma separated). If unspecified, the default is the local computer.

-Username (string)

Specifies a username of an account that has administrative permissions on the computer to add a local user group and modify the local group policy. You will be prompted for a password. Examples: `Administrator` or `TestDomain\AdminUser`.

-GroupName (string)

Specifies a name for the SAM access local user group. If unspecified, the default group name is "Secret Server Remote SAM Access"

-ForceGPUpdate

Specifies whether a group policy update should be forced for immediate effect following the script. (Otherwise Group Policy changes may take up to 120 minutes to take effect by default).

Examples

`> C:\Script\Win10-HbFix.ps1` This example gives remote SAM access to all local users on the current machine. The current PowerShell credentials would be used for authentication.

`> C:\Script\Win10-HbFix.ps1 -LogDir "D:\Win10-HbFix\log"` This example changes the default output log path to `D:\Win10-HbFix\log` (default is `[user temp directory]\log`).

`> C:\Script\Win10-HbFix.ps1 -ComputerNames "WINSERVER","TestDomain\SOMEMACHINE" -Username "TestDomain\Administrator"` This example gives remote SAM access to all local users on the WINSERVER and TestDomain\SOMEMACHINE remote computers. The domain user "TestDomain\Administrator" credentials will be used. You would be prompted for a password.

```
> C:\Script\Win10-HbFix.ps1 -ComputerNames "D:\Win10MachineList.txt" -Username "TestDomain\Administrator"
```

This example gives remote SAM access to all local users on the remote computers listed in D:\Win10MachineList.txt (one machine per line). The domain user "TestDomain\Administrator" credentials will be used. You would be prompted for a password.

`> C:\Script\Win10-HbFix.ps1 -ComputerNames "WINSERVER" -GroupName "Secret Server Group"`

This example gives remote SAM access to all local users on the WINSERVER remote computer. The local group created will be named "Secret Server Group". Current PowerShell credentials would be used for authentication.

`> C:\Script\Win10-HbFix.ps1 -ComputerNames "WINSERVER" -ForceGPUpdate -Verbose` This example gives remote SAM access to all local users on the WINSERVER remote computer, with verbose output. The current PowerShell credentials will be used for authentication. Group policy update will be forced on WINSERVER for immediate effect.

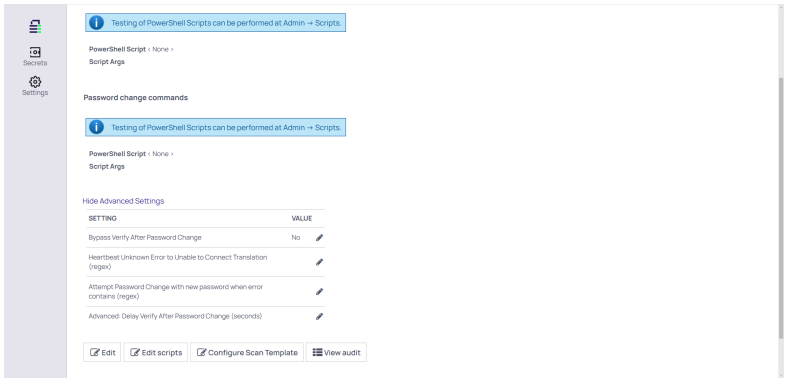
Related Articles and Resources

Network access: Restrict clients allowed to make remote calls to SAM

Option 4: Creating a Heartbeat GPO Workaround

1. Download, unzip, and run this script HBWorkAroundScripts.zip.
2. Go to **Administration > Scripts**.
3. Add the HBWorkAroundScript and the HBWorkAroundPasswordChange scripts.
4. Test the first script. Add the appropriate `args[]` as needed. Add arguments 0-4 with no quotes or commas. Spaces are the argument separator and are required.
5. You should get a return of "True," such as this:

6. Navigate to **Admin > Remote Password Changing.**
7. Click the **Options** dropdown and select **Configure Password Changers**.
8. Click the **Create Password Changer**.
9. Click the **Base Password Changer** dropdown list to select **PowerShell Script** as your password changer.
10. Type a name in the **Name** text box.
11. Click the **Save** button. The password change command page appears:



12. Click **Edit Scripts** at the bottom of the page to select the script you ran earlier.
13. Add the appropriate tokens in the **Script Args** text box.

> See "Dependency Token List" on page 1490 for a complete list.

12. Click the **Save** button.
13. Go to **Admin > Secret Templates**.

14. Select **Windows Account**.
15. Click the **Edit** button.
16. Click the **Copy Secret Template** button.
17. Click the **Configure Password Changing** button.
18. Click the **Edit** button.
19. Click the **Password Type to Use** dropdown list to select the password change you created earlier.
20. Create your windows secret using the custom template.
21. Once it is created, add your privileged and associated secret to the RPC tab as seen below. In that example we use the same one for the privileged and associated secret.
22. Run a heartbeat to confirm it works as desired.

Enabling Debug Mode in System Logs

This topic discusses enabling debug mode for application system logs for troubleshooting.

Overview

You can expand Secret Server's logging capabilities to locate additional information regarding an error or to help with troubleshooting an issue.

To enable debug mode for distributed engine log files, see "Enabling Debug Mode in DE Log Files" on page 380

Procedure

To enable debug logging mode:

1. Log in as an administrator on the application server.
2. Locate the *web-log4net.config* file. This file can be found in the web application's root directory. If you cannot locate your web application directory, see How to: Find the Web Application Root. Secret Server is typically located in the **C:\inetpub\wwwroot* directory, however, this is configurable so the location may be different in your environment.
3. Open the file in a text editor.
4. Run a find (Control+ F) command.
5. Type in *log4net* and press **Enter** to locate that section, which is usually at the top.
6. Locate the commented out *<level value="DEBUG" />*
7. Uncomment the line by removing the *<!-- and -->*.
8. Locate the *<level value="INFO" />* line in the same section.
9. Comment out the entire line by adding a *<!-- and -->* around it.
10. Restart IIS to apply the log configuration change.

This restarts all websites hosted under IIS

11. After DEBUG mode is enabled in the system log, you can reproduce the issue, investigate the error, or send the updated logs in with your support case.

Finding the Version Number of Your Secret Server Release

There are several ways to find the version number of your Secret Server release, which are detailed below.

The procedures in this section apply to the Professional and Platinum editions of the Secret Server on-premises application.

Through Secret Server

You can find the version number of your Secret Server release from any Secret Server page. Just click your user icon in the top right corner, and a box opens displaying information including the full version number.

Through Windows File Explorer

1. Log into the web server where the Secret Server application is installed.
2. Open Windows File Explorer.
3. Navigate to `C:/inetpub/wwwroot/SecretServer/bin`
4. Right-click the `Thycotic.ihawu.Business.dll` file and select **Properties**.
5. In the Properties dialog, click the **Details** tab, which displays information including your product version.

Through SQL

1. Log into your Secret Server database server.
2. Open and log into your Microsoft SQL Server Management Studio application.
3. Click **Connect**.
4. At the top level of the **Object Explorer** pane, click the **+** symbol next to servername\instance.

5. Click the **+** symbol next to **Databases**.
6. Right-click your Secret Server database and select **New Query** from the dialog.
7. In the SQL Query pane that opens, paste the following query:

```
SELECT * from tbVersion
ORDER BY CAST(REPLACE(VersionNumber, '.', '') AS INT) DESC;
```

8. Click **Execute**.
9. On the **Results** tab below the SQL Query pane, confirm that your query executed successfully and find the version number of your current Secret Server release in the top row.

Through the API

You can also use the REST API to check the version of Secret Server using the endpoint `GET /v1/version`.

Through PowerShell

1. Open a PowerShell prompt on the machine(s) running secret-server.
2. Run the following command, changing the drive and/or path as required: `(Get-Item C:\inetpub\wwwroot\SecretServer\bin\Thycotic.ihawu.Business.dll).VersionInfo.FileVersion`

Unlimited Administration Mode

Overview

Unlimited administration mode is a feature designed to allow an administrator access to all secrets and folders in their Secret Server instance without explicit permission. This can be used in the instance a company has an emergency where access to a secret is needed when no users who have permission are available. Alternately, it can be used when company policies require administrators to have access to all information in the system.

An unlimited administrator in Secret Server has extensive capabilities, including access to all secrets and folders even without explicit permission. Here are some of the key capabilities and associated risks:

Capabilities and Risks

Unlimited admin mode is a double-edged sword and must be carefully managed:

Capabilities

Unlimited administrators have:

- Complete Control: Access to all administrative features without restriction.
- Access to All Secrets: Unlimited administrators can run Secret Server in unlimited administrator mode, which grants them access to all secrets and folders.
- Audit and Reporting: Unlimited administrators can generate and view over 90 out-of-the-box reports to monitor privileged access and ensure proper password hygiene.

- Break-the-Glass Capability: This feature is part of the disaster recovery capabilities, allowing emergency access to secrets in critical situations.
- Secret Checkout Override: Unlimited administrators can access secrets even when they are checked out by another user, ensuring accountability and traceability of secret usage.
- Bypass SAML: Users with Unlimited Administrator role will effectively inherit the bypass SAML role permission and be able to bypass the SAML login process.

Risks and Mitigation

Risks

Unlimited admin mode exposes Secret Server to:

- Potential for Abuse: With the ability to access all secrets, there is a risk that an unlimited administrator could misuse their privileges, intentionally or accidentally.
- Security Gaps: Without proper monitoring and auditing, the extensive access granted to unlimited administrators could be exploited by bad actors if the administrator's credentials are compromised.
- Insider Threats: An unlimited administrator could potentially become an insider threat if they decide to act maliciously or if their account is taken over by an external attacker.

Mitigation

To mitigate these risks, it is crucial to have robust monitoring, auditing, and alerting mechanisms in place. Secret Server provides features such as automatic email alerts for unlimited-administrator-mode access, detailed audit trails, and the ability to require dual control for certain actions to enhance security.

A user with the "Unlimited Administrator" role permission can view and edit all secrets in the system, regardless of permissions—if and only if the unlimited administration mode is enabled in the configuration settings—**but** the Unlimited Administrator role does **not** have permission to enable the mode. To enable unlimited administration mode, the Administer Configuration Unlimited Admin role permission is required. This provides dual control, ensuring no single user can enable unlimited administration mode. Of course, you can bypass this safeguard by simply assigning both roles to the same user.

> The Unlimited Administrator Mode role permission is assigned to the Administrator role by default.

> A banner alert, visible to all users, displays at the top of the Secret View page when unlimited administration mode is enabled.

> The Administer Configuration Unlimited Admin was formerly called "Administer Unlimited Admin Configuration."

Enabling Unlimited Administration Mode

1. Ensure you have the Administer Configuration Unlimited Admin permission.
2. Click **Settings** on the main menu and select **Configuration Search**. The Search Configuration page appears.

3. Click the **Unlimited Admin** link. The Unlimited Admin page appears.
4. Click the **Edit** button.
5. Check to select **Enable unlimited administration mode**.
6. Add the details if prompted in the **Enter any additional notes or explanations for the configuration switch** field.
7. Click **Save**.

> We recommend administrators have specific permissions to folders and secrets and this mode is only used temporarily to assign the correct permissions.

> Changes to the administration mode are logged in an audit grid. The grid shows the user, time of the change, and any notes made by the user.

Alerts, Audits, Events, and Logs

The Alerts, Audits, Events, and Logs section provides comprehensive guidance on monitoring and managing system activity. This section is divided into four key areas: Audits, which covers audit data retention, generating audit reports, and managing PII exports; Events, which focuses on event pipelines and subscriptions for system notifications; and Logs, detailing logging procedures such as syslog configuration, SQL Server transaction logs, and setting logging levels. This section ensures users can effectively track and respond to system events, audits, and alerts to maintain security, compliance, and performance oversight.

Comparison of Secret Server Alerts, Events, Audits, and Logs

Alerts

- **Definition**: Notifications sent to users or administrators when specific actions are performed or events occur within Secret Server.
- **Customization**: Can be customized through Event Subscriptions to notify users about specific actions such as Secret Edit/Add/View, Role and Group Assignment changes, Secret expiration, Configuration changes, and Heartbeat failures.
- **Delivery**: Typically sent via email and can be configured to have high priority.
- **Purpose**: Provide real-time notifications to administrators and users about critical actions or changes, enabling them to respond promptly.

Events

- **Definition**: Specific actions or occurrences within Secret Server that are recorded for auditing and logging purposes.

- **Logging**: Events are logged and can be sent to external systems using protocols like Syslog and CEF (Common Event Format) for added security and compliance.
- **Types of Events**: Can include system events, errors, warnings, user activities, and other operational data.
- **Purpose**: Provide a detailed audit trail of activities within Secret Server, which is crucial for compliance, security monitoring, and troubleshooting.

Audits

- **Definition**: Detailed records of actions taken on secrets, including who performed the action and when it occurred.
- **Access**: The audit log for a secret can be accessed by clicking the View Audit button on the Secret View page or navigating from the User Audit report.
- **Details**: Shows the date, username, action, and any other details about the event. Includes actions like adding, updating, removing secret dependencies, editing permissions, forced expiration, and more.
- **Purpose**: Provide accountability and detailed records of changes or views on secrets, which is essential for security and compliance.

Logs

- **Definition**: Records of system events and actions that occur while Secret Server is executing.
- **Types**: Includes system logs, which can be enabled to communicate different events occurring during execution, and can be helpful in troubleshooting unexpected behavior.
- **Parameters**: System log parameters include maximum log length, notifications to administrators when the log is shrunk, and the ability to clear the log.
- **Purpose**: Provide a comprehensive record of system events for troubleshooting, monitoring, and ensuring the system is functioning as expected.

Key Differences

- **Real-Time Notification vs. Detailed Records**: Alerts provide real-time notifications, while events, audits, and logs provide detailed records of actions and system events.
- **Customization**: Alerts can be customized through Event Subscriptions, while audits and logs are automatically recorded based on system activities.
- **Purpose**: Alerts are for immediate action, events and audits are for compliance and monitoring, and logs are for troubleshooting and system monitoring.

Auditing Overview

Secret Server provides comprehensive auditing features to help organizations meet regulatory requirements and ensure security compliance. Here is an overview of the auditing capabilities in Secret Server:

Local Auditing

Secret Server locally audits all actions taken within the system.

- Auditable events include secret access, configuration changes, and user activities.
- Various user permissions are tailored for specific kinds of audits, such as viewing secret audits, user audits, and configuration audits.
- Local audit records can be accessed through the Reports tab and specific audit buttons on configuration pages.

Enhanced Auditing, Reporting, and Compliance

- Audit Reports: Generate reports to see all actions taken by a user or on a secret within a specified date range.
- Dual Control: Requires two people to access sensitive reports or recordings, enhancing security by implementing the "four eyes principle."
- Event Subscriptions: Customizable alerts that notify users or administrators when specified actions occur, such as secret edits or heartbeat failures.
- Scheduled Reports: Set up reports to be generated and sent via email on a regular schedule.
- Custom Reports: Create custom reports with database queries, including charts and rollup graphs for visualization.
- FIPS Compliance: Enable FIPS 140-2 compliant algorithms to meet U.S. Federal standards for cryptography.
- Privileged Behavior Analytics: Detect anomalies in privileged account behavior to preemptively address potential cyber threats.

Exporting and Importing Settings

- Secret Server allows exporting and importing settings, with audits recorded for each setting category that is exported or imported.
- Errors and resolutions are logged, and detailed logs are available for troubleshooting.

Alerts, Auditing, Events, and Logs

- Secret Server records specific events and can send alerts when they happen.
- Logs are maintained for various activities, providing a detailed trail of actions for auditing and compliance purposes.

Accessing Audit Records

- Local Reports: Access out-of-the-box and custom reports.
- Windows Event Log: Configure Secret Server to send audit logs to the Windows Event Log for local auditing and troubleshooting.
- Configuration Audit: View individual setting audits on the Configuration Audit page.

User Permissions for Auditing

- View Secret Audit: Allows viewing of secret audit logs.
- View User Audit Report: Allows viewing of user audit reports.

- Add Secret Custom Audit: Allows making custom audit entries via the web services API.
- User Audit Expire Secrets: Allows viewing and forcing expiration of secrets accessed by a user.

Audit Data Retention

In This Section

Overview

Secret Server can automatically delete older audit and audit-like information (both are called "audit data" here). By default, Secret Server does not delete any audit data.

Do not configure automatic record deletion for compliance or other important data.

If enabled, old data deletion occurs automatically at 0600 UTC every day. Data deletion can be run immediately by clicking the "Run Now" button. The maximum record age for each audit-data retention policy is configurable to any value greater than or equal to 30 days.

Data Retention Policies

The audit data retention offers two data retention policies:

- Personally Identifiable Information (PII): Tables containing identifiable user or organization data.
- Database Size Management: Tables that are prone to grow large, which may affect Secret Server performance.

Each policy has a title and description, which are displayed to users, as well as a defined set of Secret Server audit tables it manages. There is some overlap between the two policies' table sets as some tables fall under both PII and size management.

Personally Identifiable Information (PII)

Personally identifiable information is information such as email addresses or names that can be used to identify an individual. Some audit records contain one or more of these data types. Companies may choose to delete these records for compliance or security reasons.

The following list details which records are deleted under the PII data retention setting:

- Event Subscription Audit
- Dual Control Audit

- Group Audit
- Secret Audit
- Folder Audit
- Secret Policy Audit
- Workflow Template Audit
- Event Audit
- User audit
- Admin Log
- Access Request History
- Access Response History
- Secret access request history

Database Size Management

Certain tables in the database will grow very large over time in active enterprise organizations, which can in turn impact performance. Organizations may choose to delete these records to preserve server storage space or to prevent data-bloat and reduced performance that could accompany the growth of these tables.

The following list details which records are deleted under the database size management data retention setting:

- Group Audit
- SDK Client Audit
- Secret Audit
- Event Audit
- User audit
- Secret Log
- Secret Item Transition History
- Secret History
- User Secret Event
- Disaster Recovery Configuration Audit
- Disaster Recovery Data Replica Audit

When an audit-data retention policy runs, all records in each table for that policy that are older than the set maximum record age in days are deleted from the database. This also includes all dependent records in other tables that would otherwise prevent deletion.

Permissions

Access to the audit-data detention management pages in Secret Server is limited to users with the roles "View Data Retention" and "Administer Data Retention." As the names imply, only the latter role can manage audit data retention, such as editing and running now.

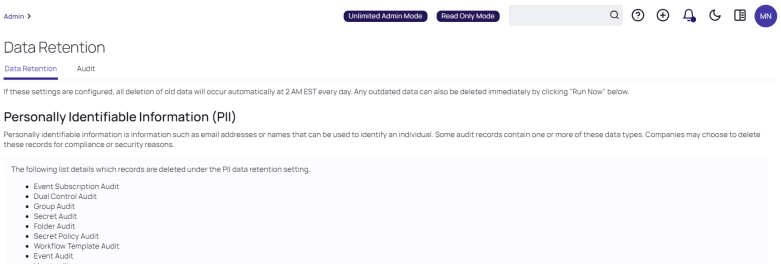
> The "Unlimited Admin" role does not include audit data retention management at this time.

By default, these two audit-data retention roles are not assigned to users. An admin must first assign the roles to users requiring access.

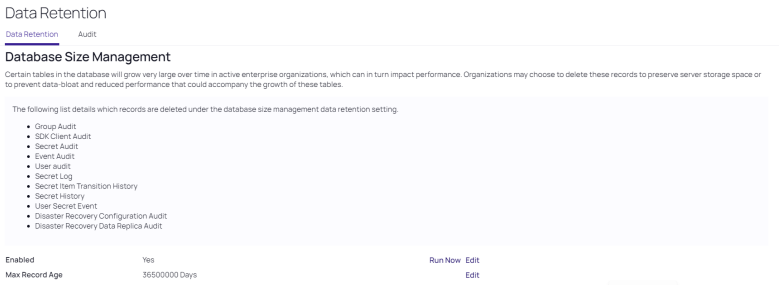
Procedures

Viewing the Status and History of Audit-Data Retention Policies

1. Go to **Admin > Data Retention**:

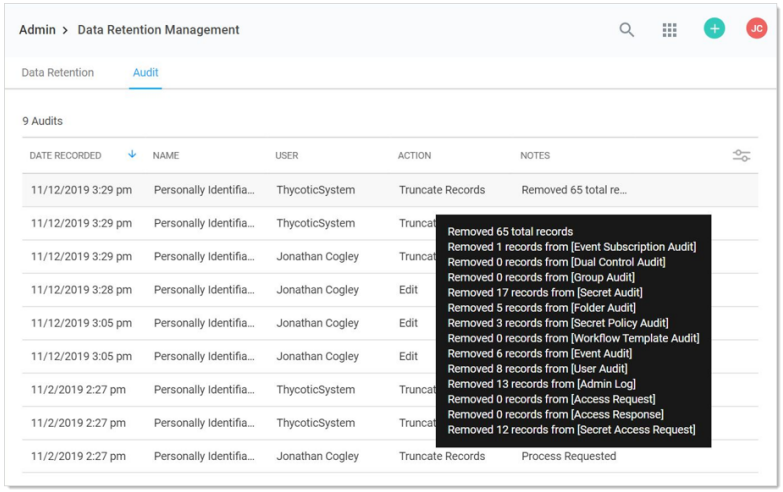


   The Personally Identifiable Information (PII) policy is displayed on the Data Retention tab. If you scroll down, you will see the Database Size policy:



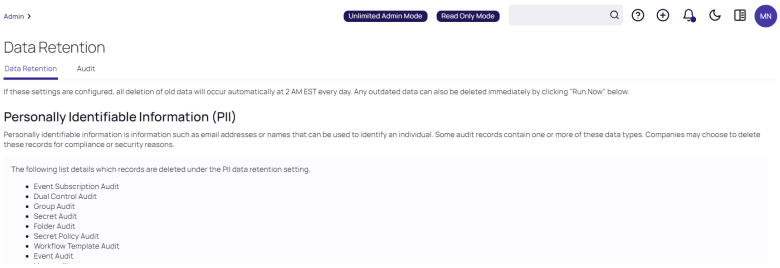
2. Notice that each policy lists:
   - The enabled status (editable)
   - The maximum age audits are allowed to remain (editable)
   - The last time the policy ran
   - The last time the policy finished running
   - All the audit data tables that the policy covers
3. To view a list of previous "runs," click the **Audit** tab. You can also hover the mouse pointer over individual

records to view details:



Editing Audit Data Policies

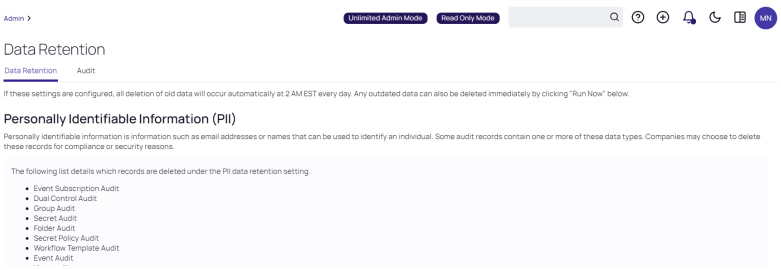
1. Go to **Admin > Data Retention**:



2. Click the **Edit** link on the **Enabled** row on the policy that you wish to edit. A popup appears (not shown).
3. Click to select the **Enabled** check box.
4. Click the **Save** button. The policy becomes enabled.
5. Click the **Edit** link on the **Max Record Age** row on the policy that you wish to edit. A popup appears (not shown).
6. Type the number of days you want to retain the data (at least 30) in the **Max Record Age** text box.
7. Click the **Save** button. The maximum record age changes.

Running an Old Audit-Data Purge Right Now

1. Go to **Admin > Data Retention**:



2. Click the **Run Now** link on the **Enabled** row on the policy that you wish to edit. A popup appears (not shown).
3. Click the **Run Now** button. The popup disappears and the policy is running now.

> If a policy is currently running and you click the Run Now button. It will not work, and a popup will tell you so. There is a built-in five-minute wait after a policy finishes before you can run it again.

4. The **Last Start Time** row changes to the current time, and a progress indicator appears.
5. When the run is complete, the **Last Complete Time** row changes to the current time.

Audit Reports

In addition to the user audit and individual secret audit, the reporting feature provides a series of activity, user, and secret reports. See "Built-in Reports" on page 883 for the most up-to-date list of reports included.

> Users can also create their own, custom reports. See "Creating and Editing Reports" on page 887.

Preventing PII Export in Audit Reports

Overview

You can enable the marking or obfuscation (hiding) of Personally Identifiable Information (PII) in audit exports. This allows for data exportation for review by third parties without including any PII. Marking PII prepares exports for external, separate cleanup, and obfuscation automatically hides or removes it during exportation.

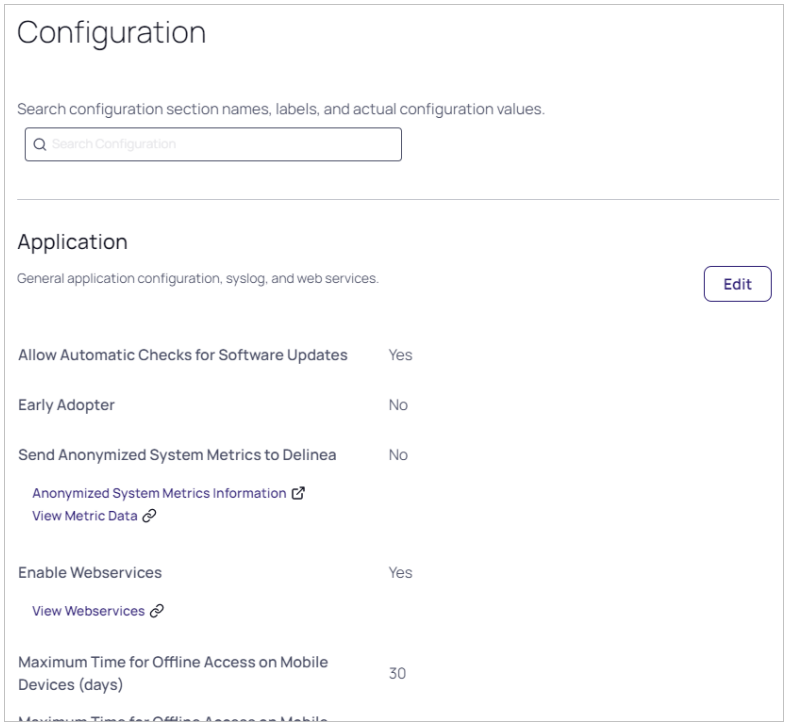
PII includes many internal stored attributes, such as IP addresses, usernames, and email addresses. Metadata fields can be flagged on creation as potentially containing PII, which aids applying the same filtering to user-configured metadata fields. PII can potentially appear in:

- System logs
- Thread logs (RPC, heartbeat, discovery, and others)
- Event logs
- Log4net files

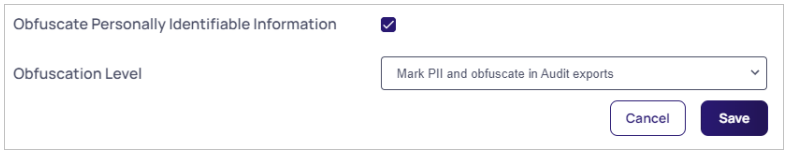
> This feature currently only applies to audit tables available in the interface.

Procedure

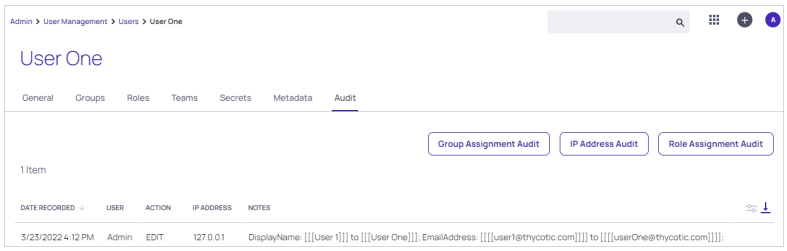
1. In the Admin Side Panel, search for and click **Application**. The Application Configuration page appears.



2. Click the **Edit** button.
3. Scroll down and click to select the **Obfuscate Personally Identifiable Information** check box. An Obfuscation Level dropdown list appears:



4. Click the Obfuscation Level dropdown list to select one of the following:
   - **Mark PII and Obfuscate in Audit Exports**: Permanently hide or remove all PII in the exported audit report. The following PII data is hashed (replaced with unique, scrambled text that provides no PII): user ID or name, IP address, and folder path. Any other PII data is replaced with `[PII REMOVED]`. The hash value allows audit viewers to see that multiple entries belong to a single user without having any specifics on who that user is.
   - **Mark PII**: Delimit all PII in easy-to-search-for markers. This allows other applications to process the PII after the export.
5. Marked PII is surrounded with three square brackets on each side, as seen below:



Obfuscated exports appear as seen in MS Excel below. Note the removed data `[PII REMOVED]` and the unique hash values for the user and display names:

Date Recorded	User	Action	IP Address	Notes
2022-03-23T20:12:10.483Z	1788267786	EDIT	1638446219	DisplayName: 665120776 to 1077913347; EmailAddress: [PII REMOVED] to [PII REMOVED];

Secret Audit Log

The audit log for a secret can be accessed by clicking the **View Audit** button on the **Secret View** page or navigating from the User Audit report. The log shows the date, the username, the action, and any other details about the event. Secret auditing provides a detailed view of each change or view on a secret.

> Audit logs are visible to anyone with the "list" permission. Thus, anybody with that permission can view permission changes, users whose permissions were changed, secret dependency information, and the machine.

Secret audits are taken for the following user actions:

- Adding, updating and removing secret dependencies
- Check out
- Editing permissions
- Forced expiration
- Hide launcher password changes
- Set for check-in
- Update
- View

For certain audit items, action notes are added providing additional details. For example, if permissions are edited, an audit record is generated detailing which users or groups gained or lost permissions. Detailed audit records add accountability to sensitive secrets where auditors or administrators need to know exactly what was modified.

Below the audit records is a **Display Password Change Log** check box. Clicking to select this check box displays logs for Heartbeat and Remote Password Changing amongst the audit items

Viewing a User Audit Report

To view a user audit report:

1. From the **Reports** page, click the **User Audit** tab.
2. From the dialog on the tab, select a user and a date range to view.
3. Click **Search History** to view the user's audit trail.

The audit search displays results for all the secrets the selected user has viewed or edited during the selected time period. The administrator has the option of expiring all the viewed secrets, to notify users to change sensitive information, or to force password changing (if the RPC is configured).

To get a full view of the actions taken on a secret, select that secret from the results list. The secret audit displays the specific user actions for a secret.

Events

The Events section is designed to help users manage and respond to system activities by providing tools and processes for monitoring important events. It includes guidance on Event Pipelines, which track the flow of system events, and Event Subscriptions, which allow users to subscribe to specific event types for timely notifications. The Notification Inbox feature is also covered, offering a centralized location for reviewing system alerts. This section ensures users can stay informed and take appropriate actions based on the events occurring within their systems.

Event Pipelines

Overview

Event pipelines (EPs) are a named group of triggers, filters, and tasks to manage events and responses to them. Event pipelines themselves can be grouped into EP policies. The Secret Server EP system is essentially a flexible instruction set builder and manager for controlling events and responses.

Event Pipeline Components

Definitions

Event Pipeline

An EP is a single named group of triggers, filters, and tasks. The same EP can be in multiple EP policies. Changing an EP affects all EP policies that EP is a part of. EPs do nothing if not assigned to an EP policy. There are two types of event pipelines–secret and user. To run an EP, include it in an active EP policy of that same type (secret or user) with set EP policy targets, such as user groups or folders.

Event Pipeline Policy

An *EPpolicy* is a named group of EPs that are run at the same time (in sequential order). Similar to EPs, there are two types of EP policies: *secret* or *user*. Secret EP policies target secret policies or folders and can only contain secret EPs. User EP policies target users in Groups and can only contain user EPs. EP policies must have an assigned EP policy target to work. Similarly, an EP policy with no assigned EPs does nothing.

Event Pipeline Filter

EP *Filters* are parameters that limit when an EP task runs. All Filters have settings and can be added to an EP multiple times. The filters are:

Secret Policy Filters

The current secret policy filters:

- Custom Variable
- Day of Week
- Event Time
- Event User: Group
- Event User: Has Two Factor
- Event User: Role
- Event User: Role Permission
- Event User: Team
- Event User: User Domain
- Event User: User Last Login
- Event User: User Setting
- IP Address
- Group
- Policy on a Secret
- Role
- Role Permission
- Secret Access Role Permission
- Secret Field
- Secret has Field
- Secret has RPC enabled
- Secret Name
- Secret Setting
- Secret Template
- Site
- Target User: Two Factor Type
- Two Factor Type

User Policy Filters

The current user policy filters:

- Custom Variable
- Day of Week

- Event Time
- Event User: Group
- Event User: Has Two Factor
- Event User: Role
- Event User: Role Permission
- Event User: Team
- Event User: User Domain
- Event User: User Last Login
- Event User: User Setting
- IP Address
- Multi-Group
- Target User: Group
- Target User: Has Two Factor
- Target User: Multi-Group
- Target User: Role
- Target User: Role Permission
- Target User: Team
- Target User: Two Factor Type
- Target User: User Domain
- Target User: User Setting
- Two Factor Type

Some filters prompt you for additional information when you select them.

Event Pipeline Policy Target

EP policy *targets* are Secret Server folders, secret policies, or user groups that are the *subject* an EP policy is applied to. For secret EP types, the secrets inside the folders or secrets under the secret policies trigger the EPs in an EP policy. As targets, folders are not recursive–only the secrets directly in the folder can trigger an EP. For user EP types, only users in the selected groups can trigger an EP.

EP policy targets are *not* the receivers of task action. Those receivers are usually components of Secret Server. The term *target* is instead used for the *subject* of an EP policy–the policy targets the secret in the policy or folder to trigger the EPs to process.

Event users are different than target users: The event user triggers the event. The target user is the recipient of the event.

Event Pipeline Task

Important: Tasks are powerful and can potentially do a lot of damage, so we highly recommend testing EPs in a safe environment before using them on production secrets.

EP *tasks* are actions that are triggered in an EP, assuming any filtering conditions are met. Tasks can edit secrets, move secrets, change permissions, send notifications, and more.

Tasks run in order of their appearance on the Task tab of the Event Pipeline details page. To change the task running order, hover the mouse pointer over the one you want to move, and use the anchor on the left of its card to drag the task to the order you want it to run. If a task fails, the follow-on tasks will not run.

> EP targets are *not* the receivers of task action. Those receivers are usually components of Secret Server. The term *target* is instead used for the *subject* of an EP policy—the policy targets the secret in the policy or folder to trigger the EPs to process.

> To reference the additional secrets in the script's Args field for the update secret with a script task or run script, use `$[ADD:1]` before the token. For example: `$[ADD:1]$USERNAME` to reference additional secret one and `$[ADD:2]$USERNAME` to reference additional secret two.)

Secret Tasks

The secret tasks are:

- Add Custom Audit
- Add Share
- Assign Secret Policy
- Assigning Site to Secret
- Change Secrets to not require a comment when viewed
- Change Secrets to not require Check Out
- Change Secrets to require Check Out
- Change Secrets to require Comment on View
- Change to Inherit Permissions
- Delete
- Disable Auto Change
- Disable Heartbeat
- Edit Share
- Enable Heartbeat
- Expire Secrets
- Fail with a message
- Hide Launcher Password

- Move to Folder
- Post Slack Message (WebHook)
- Retry with new random password
- Run Heartbeat
- Run Script
- Schedule Pipeline
- Secret: Add Custom Audit
- Secret: Add Share
- Secret: Assign Secret Policy
- Secret: Assigning Site to Secret
- Secret: Change Password Remotely
- Secret: Change Secrets to not require a comment when viewed
- Secret: Change Secrets to not require Check Out
- Secret: Change Secrets to require Check Out
- Secret: Change Secrets to require Comment on View
- Secret: Change to Inherit Permissions
- Secret: Delete
- Secret: Disable Auto Change on Secret
- Secret: Disable Heartbeat
- Secret: Edit Share
- Secret: Enable Auto Change on Secret
- Secret: Enable Heartbeat
- Secret: Expire Secrets
- Secret: Fail with a message
- Secret: Move to Folder
- Secret: Retry with new random password
- Secret: Run Heartbeat
- Secret: Send Email to Owners
- Secret: Set Privileged Account
- Secret: Stop RPC
- Secret: Undelete
- Secret: Update Secret by field
- Secret: Update Secret Name

- Secret: Update Secret with a script
- Secret: Viewing Password Does Not Require Edit
- Secret: Viewing Password Requires Edit
- Send Email to Event User
- Send Email to Group
- Send Email to List
- Send Email to Owners
- Set Custom Variable
- Set Privileged Account
- Stop RPC
- Undelete
- Unhide Launcher Password
- Update Secret by field
- Update Secret Name
- Update Secret with a script
- Update Secrets to automatically change the password

User Tasks

The user tasks are:

- Post Slack Message (WebHook)
- Run Script
- Schedule Pipeline
- Send Email to Event User
- Send Email to Group
- Send Email to List
- Set Custom Variable
- Target User: Add User to Group
- Target User: Add User to Team
- Target User: Disable Duo Two Factor
- Target User: Disable Email Two Factor
- Target User: Disable FIDO2 Two Factor
- Target User: Disable RADIUS Two Factor
- Target User: Disable TOTP Auth Two Factor
- Target User: Disable Users

- Target User: Enable Duo Two Factor
- Target User: Enable Email Two Factor
- Target User: Enable FIDO2 Two Factor
- Target User: Enable RADIUS Two Factor
- Target User: Enable TOTP Auth Two Factor
- Target User: Enable Users
- Target User: Force Logout
- Target User: Lock User
- Target User: Remove User from Group
- Target User: Remove User from Team
- Target User: Reset FIDO2 Two Factor
- Target User: Reset TOTP Auth Two Factor
- Target User: Send Email to Target User
- Target User: Unlock User

Event User

An event user is the user making the action. For example: Admin updated user Jane's email. Admin is the event user.

Event Variable

An event variable is a place holder for a piece of information that will manifest when the event occurs, for example the user initiating the event ($ByUser) or whether or not the applicable secret is active ($secret.active).

Target User

A target user is the affected user. Example: Admin updated user Jane's email. Jane is the target user.

Triggers

EP *triggers* are events in Secret Server that cause the EP to begin processing. All triggers have no settings and can only be added to an EP once.

Pipelines are triggered following an action in most cases—pre-checkout being the exception. Therefore, you cannot interrupt an action during an event.

The triggers are:

Secret Triggers

- Access Approved
- Access Denied

- Cache View
- Check In
- Check Out
- Copy
- Create
- Custom Audit
- Custom Password Requirement Added To Field
- Custom Password Requirement Removed From Field
- Delete
- Dependency Added
- Dependency Deleted
- Dependency Failure
- Edit
- Expired Today
- Expires in 1 Day
- Expires in 15 Days
- Expires in 3 Days
- Expires in 30 Days
- Expires in 45 Days
- Expires in 60 Days
- Expires in 7 Days
- Export
- File Save
- Heartbeat Failure
- Heartbeat Success
- Hook Create
- Hook Delete
- Hook Edit
- Hook Failure
- Hook Success
- Launch
- Password Change
- Password Change Failed

- Password Change Maximum Attempts Reached
- Password Displayed
- Pre-Check In

> When using the Pre-Check In trigger, we recommend applying a group filter too. That trigger is a blocking call prior to secret check in that runs a script or causes the check in to fail with a warning. A problem arises when Secret Server does the same check-in process for the system "user" in the background at the end of the checkout interval. When the Pre-Check In trigger causes the check in to fail with a warning, the Secret Server background process continues to attempt check in forever, causing Secret Server to disable the pipeline. Applying a group filter ensures the trigger does not apply to the system user.

- Pre-Check Out
- Secret Policy Change
- Session Recording View
- Undelete
- View
- Viewed Secret Edit
- Web Password Fill

User Triggers

- Added to Group
- Challenge Applied
- Challenge Cleared
- Disable
- Enable
- Lockout
- Login
- Login Failure
- Logout
- Owners Modified
- Remove Personally Identifiable Information
- Removed From Group
- Two Factor Changed
- Two Factor Reset Failure
- Two Factor Reset Success
- User: Create

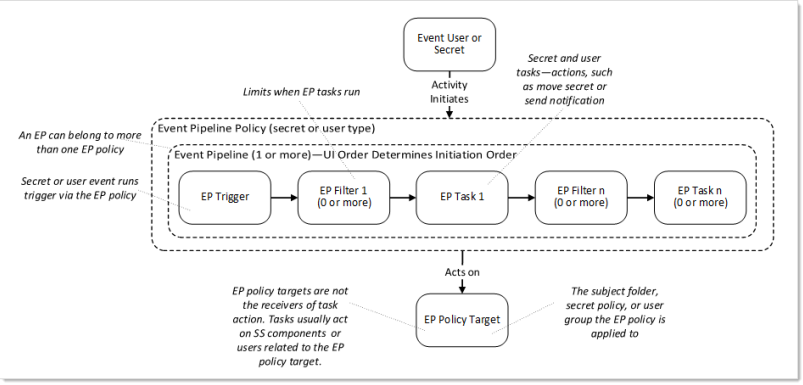
- User: Edit
- User: Password Change

Component Relationships

The following diagram shows how the components in the Definitions section relate.

Please refer to the Definition section when viewing this diagram.

Figure: Component Relationships



Event Variables

Event variables are used in EP filters or tasks. They are:

Secret Field Tokens

These can be any secret field name in the tbSecretField table that is not a Password (IsPassword=0) or File (IsFile=0) type. For example, for an Active Directory Account (SecretTypeID=6001), these tokens are available: $Username, $Domain, or $Notes.

Event Setting Tokens

Table: Event Setting Tokens with Filter Values

- $metadata.secret.section.fieldname
- $metadata.folder.section.fieldname
- $metadata.user.section.fieldname
- $metadata.secretfolderuser.section.fieldname

Token Name	Purpose	Values
$ByUser	Username that initiated the event	Text

Token Name	Purpose	Values
$ByUserDisplayName	Display name of user that initiated event	Text
$ContainerName	Folder name for the event	Text
$EventAction	Action that occurred on the event entity type. See list of triggers.	Text
$EventDetails	Event notes. For heartbeats and RPC, this contains the status and any error message.	Text
$EventUserKnownAs	Username for user that caused the event. If a domain account exists, then this appears as domain\username.	Text
$ItemId	Secret ID for the event	Text
$ItemNameForDisplay	Event secret name	Text

Secret Setting Tokens

Table: Secret Setting Tokens with Filter Values

Token Name	Purpose	Values
$Secret.Active	Active	Boolean
$Secret.AutoChangeOnExpiration	Auto change on expiration	Boolean
$Secret.ChangePasswordNow	Change password now	Boolean
$Secret.CheckOutChangePassword	Checkout change password	Boolean
$Secret.CheckOutEnabled	Checkout enabled	Boolean
$Secret.EnableInheritPermissions	Enable inherit permissions	Boolean
$Secret.EnableInheritSecretPolicy	Enable inherit secret policy	Boolean

Token Name	Purpose	Values
$Secret.Expired	Expired	Boolean
$Secret.HideLauncherPassword	Hide launcher password	Boolean
$Secret.IsDoubleLock	Double lock	Boolean
$Secret.IsSessionRecordingEnabled	Session recording enabled	Boolean
$Secret.IsSSHProxyEnabled	SSH proxy enabled	Boolean
$Secret.LastHeartBeatStatus	Status of last heartbeat	AccessDenied; AccountLockedOut; ArgumentError; Disabled; DnsMismatch; Failed; IncompatibleHost; Pending; Processing; Success; UnableToConnect; UnableToValidateServerPublicKey; UnknownError
$Secret.PasswordChangeFailed	Password change failed	Boolean
$Secret.PasswordChangeOutOfSync	Password change out of sync	Boolean
$Secret.PasswordChangeStatus	Password change status	None; Pending; Processing
$Secret.PasswordComplianceCode	Password compliance code	Pending; Pass; Fail
$Secret.RequireApprovalForAccess	Require approval for access	Boolean

Token Name	Purpose	Values
$Secret.RequireApprovalForAccessForEditors	Require approval for access for editors	Boolean
$Secret.RequireApprovalForAccessForOwnersAndApprovers	Require approval for access for owners and approvers	Boolean
$Secret.RequireViewComment	Require view comment	Boolean
$Secret.RestrictSshCommands	Restrict SSH commands	Boolean
$Secret.RPCAttemptCount	RPC attempt count	Boolean
$Secret.SecretId	Secret ID	Text
$Secret.SecretPolicyId	Secret policy ID	Text
$Secret.SecretTemplateName	Secret template name	Text

Additional Tokens

Secret

- $SecretName
- $SecretId

Folder

- $FolderId
- $FolderName
- $FolderPath

Event User

- $EventUserDomain
- $EventUserKnownAs
- $EventUserName
- $EventUserLastLogin
- $EventUserId

Metadata

- $metadata.folder.section.fieldname (secret event pipelines)
- $metadata.secret.section.fieldname (secret event pipelines)
- $metadata.secretfolderuser.section.fieldname (secret event pipelines)
- $metadata.user.section.fieldname (secret and user event pipelines)
- $metadata.user.section.fieldname (user event pipelines)

Target User

- $TargetUser.DisplayName
- $TargetUser.IsApplicationAccount
- $TargetUser.IsSystemUser
- $TargetUser.UserEmail
- $TargetUser.UserEnabled
- $TargetUser.UserName
- $TargetUserDomain
- $TargetUserId
- $TargetUserKnownAs
- $TargetUserLastLogin
- $TargetUserName

Custom Task Variables

These are variables created with the EP task. There are two types, global and item, both of which are referenced in the same way.

You must set a custom variable before using it. Thus, you cannot set a variable and use it in the same pipeline. One way around this is to create two pipelines in a policy–the first pipeline sets the variable and the second one uses it. Another way is to first set the variable in Secret Server.

Global Variable

- $GlobalVariable.CustomVariableName
- This custom task variable is global, so there should only be one per variable name.

Item Variable

- $ItemVariable.CustomVariableName
- This variable is per SecretId (secret pipeline) or UserId (user pipeline).

> The first time an EP task is invoked, an item variable is not translated, but subsequent invocations have the variable. Global variables are immediately available.

Permissions

There are three permissions:

- **Administer Pipelines:** Allows the user to create, edit, and remove EPs and EP policies.
- **Assign Pipelines:** Allows the user to assign an EP policy to secret policies, or folders.
- **View Pipelines:** Allows the user to view EP policies and policy activities.

Procedures

Event Pipelines

Activating or Deactivating Event Pipelines

To control if an EP is available to all EP policies, you can toggle the EP's active status:

1. Go to the **Event Pipelines** page.
2. Click the **Pipelines** tab.
3. Locate the card for the EP you want to activate or deactivate.
4. Click the **Active/Inactive** toggle button. A confirmation popup appears.
5. Click the **OK** button. The EP's status is changed for all EP policies it belongs to.

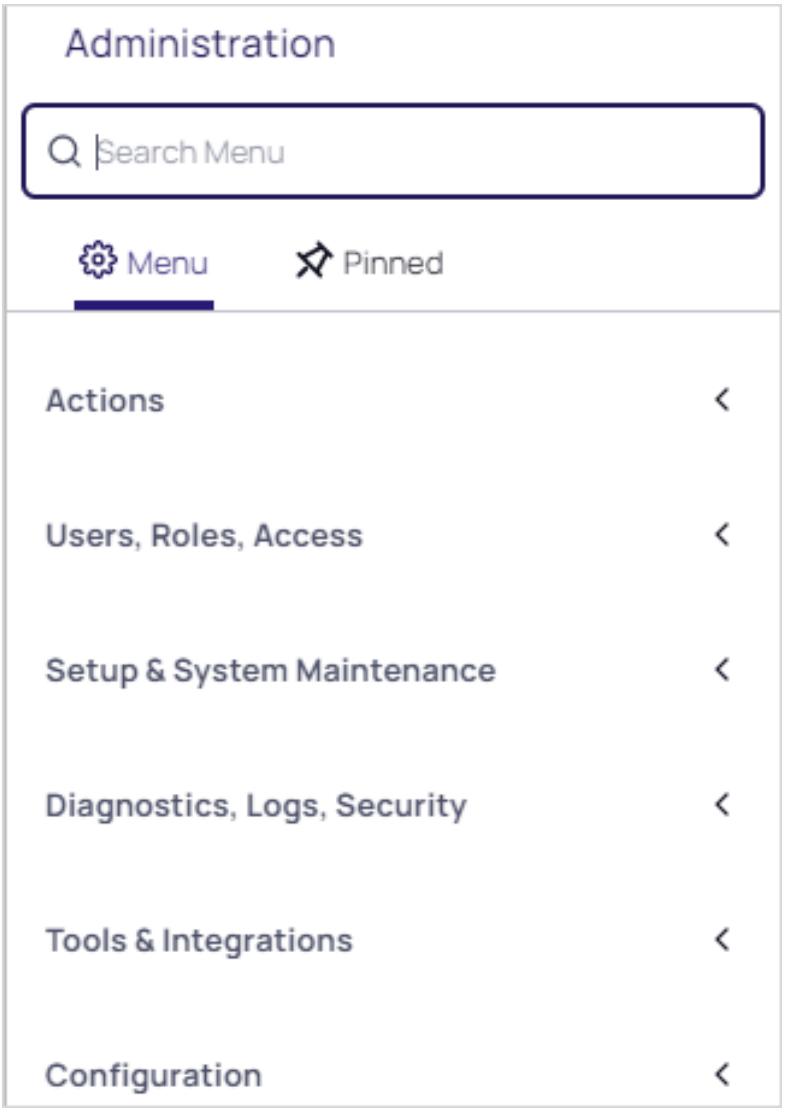
Creating New Event Pipelines

> You can create EPs from the Event Pipelines list (shown below) or an EP policy's details view. With the former method, you will have to add the EP to an EP policy separately. With the latter method, the EP is automatically added to the EP policy you are viewing. You can later manually add additional EPs to the policy as desired.

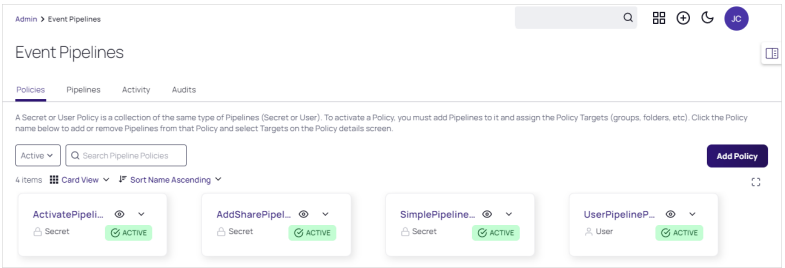
To create a new EP:

Step One: Create the EP

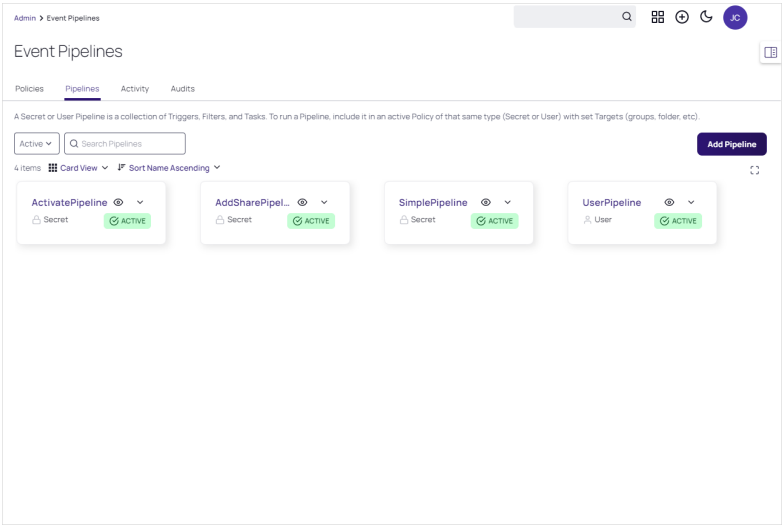
1. Navigate to **Administration**. The Admin Side Panel appears:



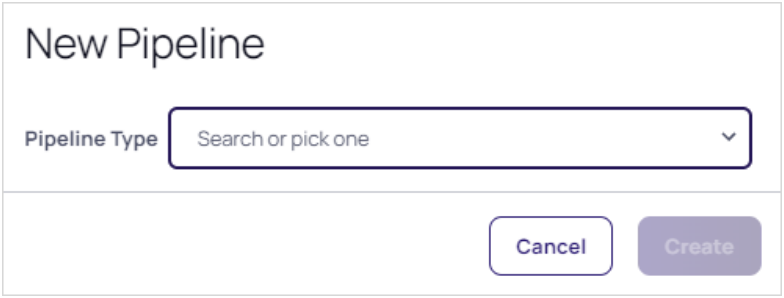
2. Click **Actions** and select **Event Pipeline Policy** in the list. The Event Pipelines: Policies tab of the Event Pipelines page appears:



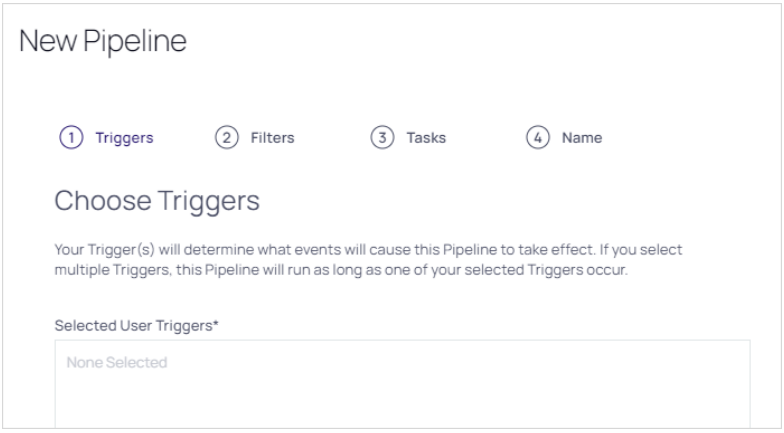
3. Click the **Pipelines** tab:



4. Click the **Add Pipeline** button. The New Pipeline popup appears:



5. Click the **Pipeline Type** dropdown list to select the EP type: **Secret** or **User**. For this instruction, we chose User.
6. Click the **Create** button. The New Pipeline wizard appears on the Choose Triggers page:

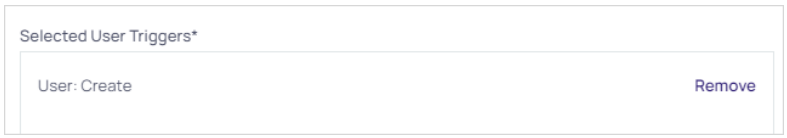


Step Two: Add Triggers

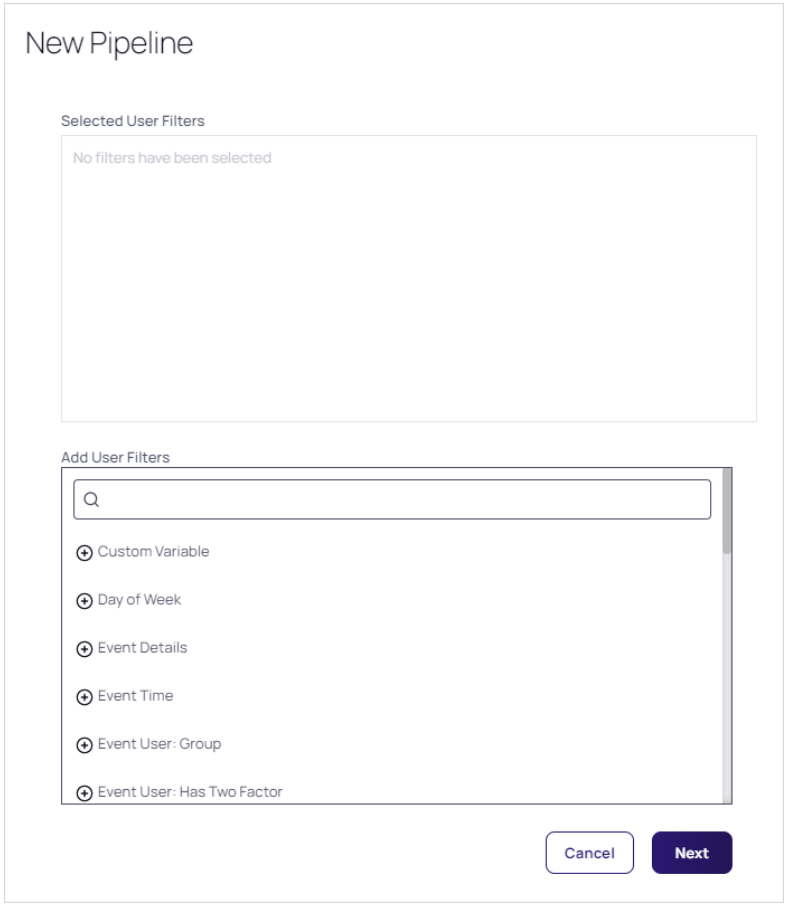
> When using the pre-check-in trigger, we recommend applying a group filter too. That trigger is a blocking call prior to secret check in that runs a script or causes the check in to fail with a warning. A problem arises when Secret Server does the same check-in process for the system "user" in the background at the end of the checkout interval. When the Pre-Check In trigger causes the check in to fail with a warning, the Secret Server background process continues to attempt check in forever, causing Secret Server to disable the pipeline. Applying a group filter ensures the trigger does not apply to the system user.

1. In the **Add Triggers** section, click the **+** button next to the triggers you desire. You can also search for a trigger by typing in the search text box. The selected triggers appear in the Selected Triggers list. Consider the following when selecting triggers:
   - Currently triggers are centralized around events that are linked to a secret.
   - You can add multiple triggers.
   - You can limit when the EP runs by adding filters.
   - Multiple triggers are logically ORed (not XORed) together. Each trigger is considered individually, and only one needs to apply for the EP to run—if concurrent triggers do not apply, it does not matter. If multiple triggers do apply, the EP will only run once per EP policy.

   The added trigger appears in the Selected User Triggers box:



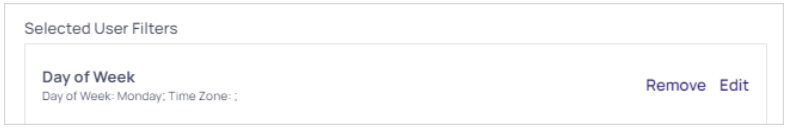
2. Click the **Next** button. The Choose Filters page of the wizard appears:



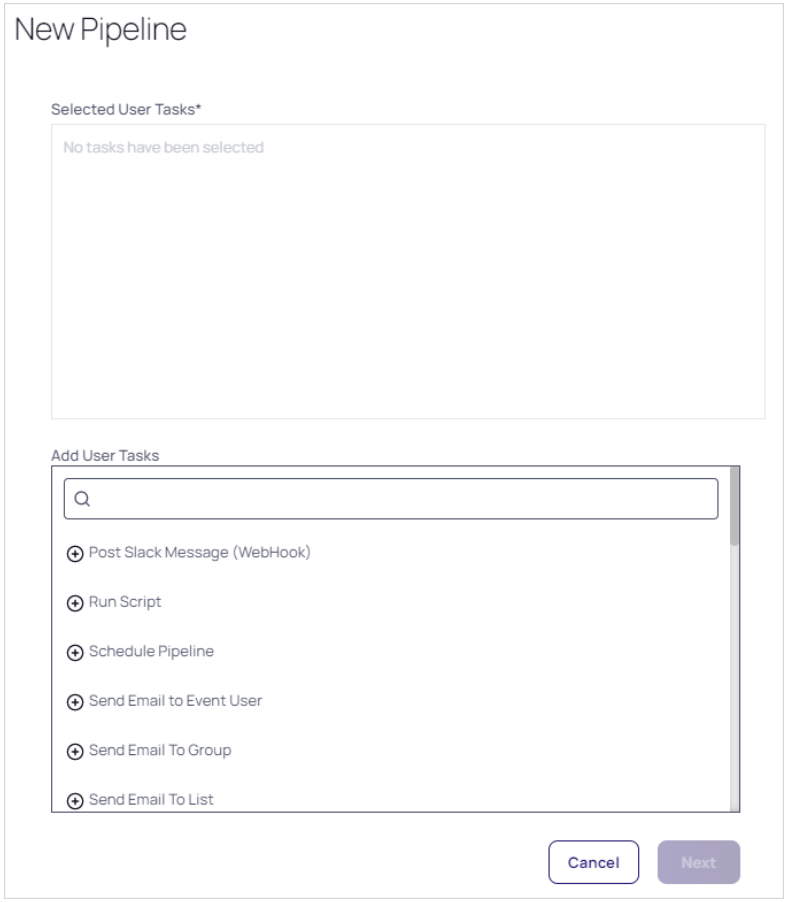
Step Three: Add Filters

1. Use the exact same method to add filters to the EP. All filters present a popup page for you to provide additional information when you click on them. Consider the following when selecting filters:
   - Whereas triggers focused on secrets, filters can also access secret and user information.
   - Because the same filter can differ by its settings, you can add the same filter multiple times to an EP.
   - Filters are logically ANDed together–all filters apply at once and all matter.
   - Metadata (MetaData) filters are available for secrets, folders, or user (secret/folder/user hierarchy). Metadata filters can filter for group names.

   The selected filters appear in the Selected User Filters section:

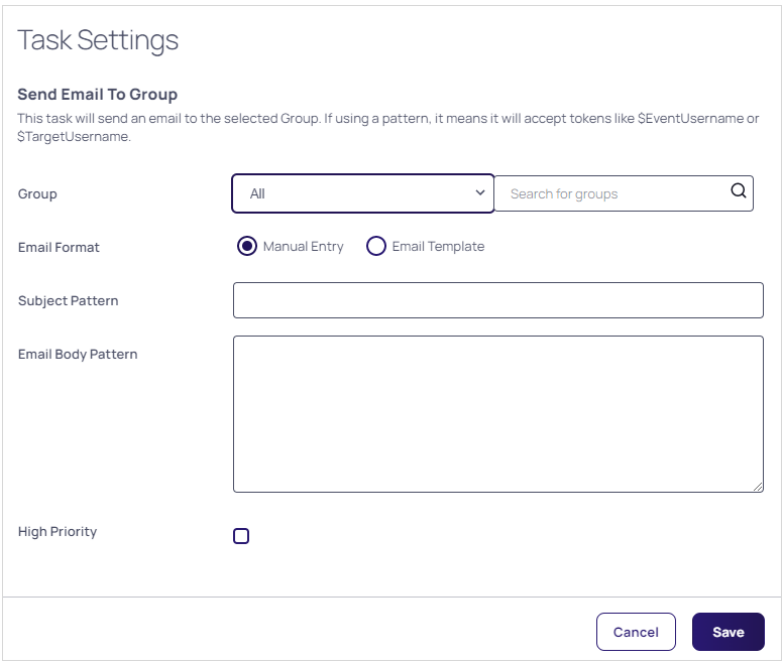


2. Click the **Next** button. The Choose Tasks page of the wizard appears:



Step Four: Choose Tasks

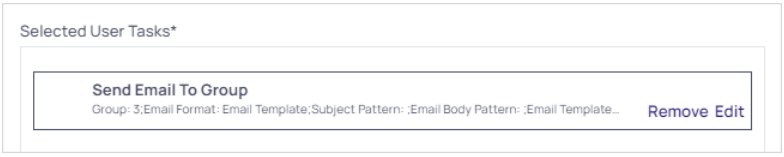
1. Use the exact same method to add tasks to the EP. Many tasks present a popup page for you to provide additional information when you click on them. For example, we chose Send Email to Group:



In this instance, you would choose the groups and the email format, either manual or using an email template, which is the same as an inbox template (see "Using Inbox Templates" on page 334).

> The email templates for EPs can use the EP tokens in addition to the standard inbox template tokens in the message subject and body. Messages are sent directly to the user's inbox based on the email addresses generated by the send email task, not subscribers from any inbox rule.

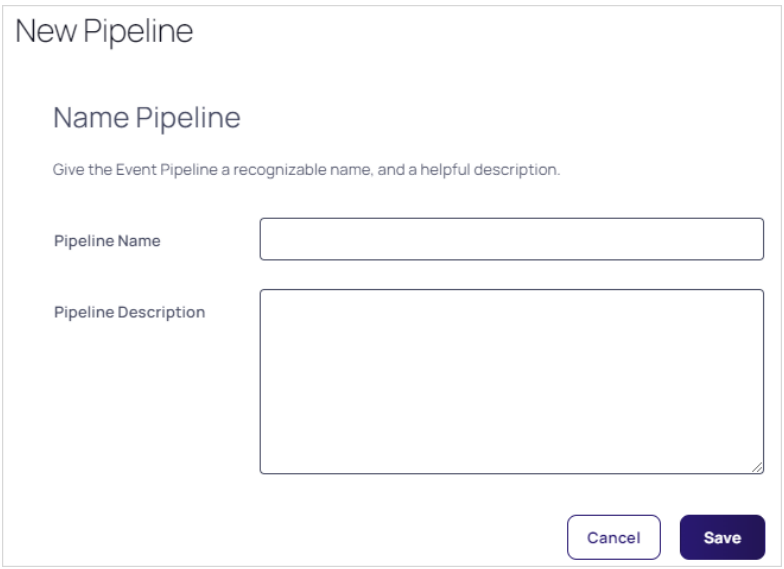
2. The selected user tasks appear :



3. Set the task order if you selected more than one. Tasks run in order of their appearance in the **Task** tab of the **Event Pipeline** page. To change the task running order, hover the mouse pointer over the one you want to move, and use the anchor on the left of its card to drag the task to the order you wish it to run. If a task fails, then the following tasks will not run.

> Tasks are very powerful and thus can be dangerous. You can alter Secret Server in dramatic, sometimes irreversible ways. We strongly recommend testing EPs in a safe sandbox environment before applying them to production Secret Server servers.

4. Click the **Next** button. The Name Pipeline page of the wizard appears:



5. Type the EP's name in the **Pipeline Name** text box.
6. Type a description of the EP in the **Pipeline Description** text box.
7. Click the **Save** button.

Editing Existing Event Pipelines

To create an EP:

1. Go to the **Event Pipeline** page.
2. If necessary, click the **Pipelines** tab. The Event Pipeline Pipelines page appears.
3. Click the title of the card representing the EP you want to edit. The EP wizard appears.
4. See Creating New Event Pipelines for instructions on using the wizard.

Viewing Event Pipelines

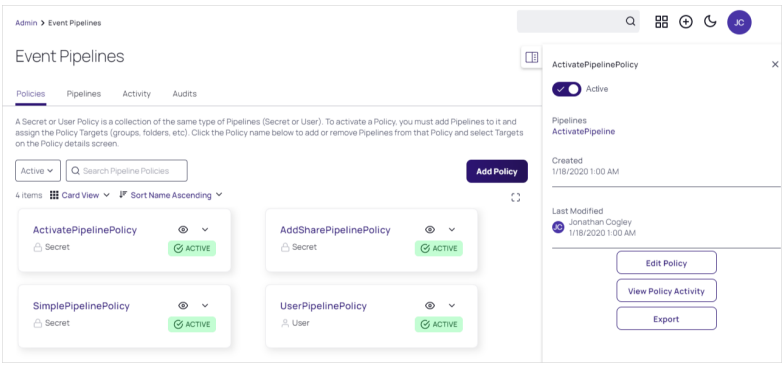
Because EPs are not directly tied to a single EP policy, they can be viewed through an EP policy or directly from the EP list. The EP list is a tab on the main Event Pipeline Policy page directly after navigating from the Admin page. After selecting an EP policy, its associated EPs are displayed in cards.

Event Pipeline Policies

Activating or Deactivating Event Pipeline Policies

To control if an EP policy is available, you can toggle its active status:

1. Go to the **Event Pipelines** page.
2. If necessary, click the **Policies** tab.
3. Locate the card for the EP policy you want to activate or deactivate.
4. Click the eye icon to open the preview panel:



5. Click the **Active/Inactive** toggle button. A confirmation popup appears.
6. Click the **OK** button. The EP policy's status is changed.

> The EPs belonging to the EP policy remain available to other EP policies.

Adding an Existing Event Pipeline

> Adding an existing pipeline enables that pipeline to be used in other policies. Only pipelines of the same type (secret or user) can be added.

> This does not create a copy of the existing pipeline, it creates a link. Thus, any changes to the pipeline will affect the other policies that use it.

1. Go to the **Event Pipelines** page.
2. If necessary, click the **Policies** tab.
3. Select the EP policy you want to add a pipeline to.
4. Click the **Add Pipeline** button.
5. Click the **Add Existing Pipeline** dropdown list and select the pipeline (only pipelines of the same type will show).
6. Click the **Create** button.

Assigning Folders and Secret Policies to Event Policy Targets

Folders

1. Go to the **Event Pipeline** page.
2. If necessary, click the **Policies** tab. The Event Pipeline Policies page appears.
3. Click the title of the EP policy on its card on the **Event Pipeline Policies** page. The page for that EP policy appears.
4. Click the **No Folder Selected** link in the **Targets** section. A destination page appears.

5. Click to select the check boxes for the desired target folders in the tree. Click the tiny arrow next to the check box to expand the tree. Remember, selecting a folder does *not* automatically select its subfolders.
6. Click the **Save** button.

Secret Policies

1. Click **Admin > Secret Policies**. The Secret Policy page appears.
2. Click the desired secret policy's name in the list. The Secret Policy page for that policy appears.
3. Click the **Edit** button. The list becomes editable.
4. Click the **Event Pipeline Policy** dropdown list in the **Security Setting** section and select **Enforced**.
5. Click the **Save** button. All secrets under that secret policy are now affected by the EP policy.

Creating, Importing, and Duplicating Event Pipeline Policies


Newly added EP policies are deactivated by default.

1. Click **Admin > See All**.
2. Click the **Action** button and select **Event Pipeline Policy**.
3. If you plan to duplicate an existing EP policy, click the card for that policy in the **Event Pipeline Policies** list.
4. Click the **Add Policy** button, and you will be presented with the following options:
   - **Create New Policy:** Click the selection button, and type a name in the **Policy Name** text box, and optionally type a description in the **Policy Description** text box.
   - **Import Policy:** Import an exported EP policy in JSON format. This can be a policy exported from a separate Secret Server instance. Click the selection button, and paste the JSON payload in the **Add Policy** text box, click the **Create** button.
   - **Duplicate Selected Policy:** Copy an existing EP policy. Click the selection button, and then click the **Create** button. The new EP appears in the Event Pipeline Policies list.

Monitoring Event Pipeline Policies

There are two ways to monitor your EP policy:

- **Audit:** Shows changes to EP policies, targets, and EPs. Click the **Audits** tab on the **Event Pipeline Policies** page.
- **Activity:** Shows the actions each EP policy or single EP took each time it is triggered. This includes failures, skips, and successes. Click the card for the desired EP policy, and then click the **View Policy Activity** button on the right. Alternatively, you can click the title on the card. When the page for the EP policy appears, click the **Activity** tab.

Ordering Event Pipelines in Event Pipeline Policies

Event Pipelines run in order they appear in the EP policy. Since EPs can be in multiple EP policies, the order is unique to each policy. To change the EP order in the EP policy:

1. Go to the **Event Pipeline Policies** page.
2. Click the name on the card for the EP policy you want to edit. The policy's page appears on the Details tab.
3. Hover the mouse pointer over the EP you want to reorder. An anchor appears on the left of the card.
4. Drag that anchor to the desired position.

> If an error occurs in a policy's EP, then the following EP still runs.

Removing Event Pipelines from Event Pipeline Policies

To remove an EP from an EP:

1. Go to the **Event Pipeline Policies** page.
2. Click the name on the card for the EP policy you want to edit. The policy's page appears on the Details tab.
3. Click on an EP in the details of an EP policy. A panel appears on the right of the page.
4. Click the **Remove Pipeline** button.

> The button removes the EP from the EP policy, but it does not remove it from Secret Server. Other EP policies using the EP still have access to it.

Advanced Settings and Troubleshooting

Configuring Advanced Settings

There are a few new advanced settings you can use with EP polices:

- **Event Pipeline Activity Log entries removed after (days)**: The EP activity log entries stay in the log for this many days. Default value: 90.
- **Event Pipelines: Allow Confidential Secret Fields to be used in Scripts**: Allows confidential secret fields to be used in EP script, such as $password. Default value: False.
- **Event Pipelines Infinite Loop Time (Minutes)**: If an EP executes the number of times specified in the infinite loop threshold during the Infinite Loop Time period, it is marked as an infinite loop. Default Value: 5 (on premises), 20 (cloud).
- **Event Pipelines Infinite Loop Threshold:** Number of times that an EP can execute within the infinite loop time on an individual item before it is considered to be an infinite loop. Default Value: 5.
- **Event Pipelines Log Skipped Policies**: If true, the pipeline activity log will log filtered policies runs. Default value: False.
- **Event Pipelines Maximum Script Run Time (Minutes)**: Scripts ran by EP tasks are stopped after this many minutes. Default Value: 5 minutes.
- **Heartbeat: Include UnableToConnect as Heartbeat Failure Event**: Adds the ability to trigger EPs on heartbeat UnableToConnect status. When toggled to true, this setting allows the user to include UnableToConnect as part of the heartbeat failure EPs. It defaults to false.

Infinite Loops

It is possible for EPs to trigger each other over and over in an endless loop. For example:

1. Editing a secret triggers one EP to run a heartbeat on the secret.
2. The heartbeat triggers another EP to edit the secret.
3. Editing the secret triggers the original EP to run another heartbeat, restarting the cycle, creating an infinite feedback loop.

Fortunately, Secret Server detects these loops and automatically deactivates the involved EPs. So, if you have EPs that seem to be deactivating themselves, look for circular logic paths involving the EPs.

> By default, pipelines are configured to consider any event that executes five tasks within five minutes from the same trigger as an infinite loop. For example, "secret edit" is selected as a pipeline trigger, and "remote password change" is selected as the task. After the first edit is made on a secret, an RPC is triggered. Every time the RPC completes, a new edit is triggered, which, in turn, triggers another RPC. If this happens five times within five minutes, then an infinite loop is declared. If the RPC is slow, taking more than five minutes for five password changes to occur, then an infinite loop is **not** declared. In this case, use the "configuration advanced" page to change "event pipelines infinite loop time (minutes)" to a longer time.

Event Subscription Overview

Event Subscriptions in Secret Server are a powerful feature designed to keep administrators and users informed about critical activities and changes within the system. These subscriptions trigger notifications for defined events, ensuring that relevant stakeholders are promptly alerted to important occurrences.

Customizable Alerts

Event subscriptions can be set up to alert users or administrators when specific actions are performed or events occur.

Notifications can be sent to the inbox and further communicated externally via email or Slack.

Event Types

- Unlimited Administration Mode toggle
- Secret Edit/Add/View
- Role and Group Assignment changes
- Secret expiration
- Configuration changes
- Heartbeat failure when a password is invalid.

Notification Management

- Notifications are an alert of specific events and not intended for archived reporting.
- Users can manage how long notifications last in the inbox before expiration.
- High-priority emails can be sent for critical events.

User and Group Subscriptions

- Event subscriptions can be assigned to specific users or groups within Secret Server.
- This ensures that only the relevant individuals receive notifications about events that concern them.

Example Use Cases

- Security Alerts: Notify administrators immediately when there is a configuration change or a heartbeat failure, ensuring quick response to potential security issues.
- Compliance Monitoring: Keep track of role and group assignment changes to maintain compliance with internal policies and external regulations.
- Operational Efficiency: Alert IT teams about secret expirations or edits to ensure smooth and secure operations.

Creating Event Subscriptions

Event subscriptions trigger notifications of defined events within the system. These notifications are sent to the inbox, which may send them externally via email or Slack, depending on your configuration.

These notifications are an alert of specific events and not intended to be used for archived reporting.

To add an event subscription:

Task 1: Creating an Event Subscription

1. Navigate to **Admin > Event Subscriptions**.
2. Click the **Create Event Subscription** button. The Create Event Subscription page appears.
3. In the **Name** text box, enter a name for this new event subscription.
4. Click to select the **Send Email** check box if you want to send an email via an inbox notification.
5. Click to select the **Send Slack** check box if you want to send an Slack message via an inbox notification.

Task 2: Adding Events

Create the events that trigger notifications:

1. Click the **Events** dropdown list to select an event object to trigger a notification.
2. Click the second **Events** dropdown list to select the event for the chosen object. For some events, one or more follow-on dropdown lists may appear. For example, if you chose Secret and then Create, another dropdown list would appear for you to select whether you want all secrets, those in the selected folder or those in the selected

folder and its subfolders. Similarly, if you chose those in a folder, a link appears for you to chose the folder.

3. Click the **Add Event** button to add the event. An event table appears below.
4. Add more events as desired.
5. Click the **Create Event Subscription** button. The event subscription's page appears.
6. If you want to change the status from active to inactive or adjust the number of days before an event notification expires, click the **Edit** link next to **Event Subscriptions** and make changes.

Task 3: Adding Subscribers

Subscribers are users that are explicitly defined or are a subscribed group member. They receive a notification in their inbox when this event subscription is triggered.

> Communication to external emails or other channels is defined by inbox notification rules.

1. Scroll down the **Subscribers** section and click **Edit**.
2. Click the **Add** button. The Users & Groups popup page appears.
3. Type the name of the user or group you want to add in the **Search** text box.
4. Click to select the check box next to the user or group that remains.
5. Repeat the process for additional users or groups.
6. Click the **Add** button. The new subscriber appears in a table in the section.

Task 4: Associating Rules

Inbox rules filter a specified event subscription (or other alerts)—in this case, the one you just created. When you create an event subscription, an inbox rule based on the event subscription system rule is automatically created. It can be updated or reviewed to change the actions, email template, or other communication preferences.

The event subscription subscribers defines who *potentially* receives the event alert. The associated inbox rules filters which events are shared with those subscribers via Slack or email messages. Inbox rules search for specified text strings in specified locations in the incoming notification.

> Notifications can be triggered by defined conditions and not a specific event subscription. Those rules may not appear in this list.

1. Scroll down to the **Associated Rules** section. Note that a link to your new event subscription inbox rule already appears at the bottom.
2. If you want to edit the rule, click the link for the inbox rule. See "Using Inbox Rules" on page 321.

Disabling an Event Subscription

To disable an event subscription:

1. Navigate to **Admin > Event Subscriptions** and click the subscription name link.
2. Click to uncheck the **Active** checkbox on the following page.

Editing an Event Subscription

To edit an event subscription:

1. Navigate to **Admin > Event Subscriptions**, click the subscription name, and then **Edit**.
2. To remove a subscribed user, group, or event, click the button next to the entry in the appropriate list.
3. To add entries to either list, see "Creating Event Subscriptions" on page 311.
4. Click **Save** to save all changes.

Event List

The following events are available:

Table: Folder Events

Event	Scope
Create	All, In this Folder
Delete	All, For this Folder, In this Folder
Edit Permissions	All, For this Folder, In this Folder
Secret Policy Change	All, For this Folder, In this Folder

Table: Secret Events

Event	Scope
Access Approved	All, For this Secret, In this Folder
Access Denied	All, For this Secret, In this Folder
Cache View	All, For this Secret, In this Folder
Check In	All, For this Secret, In this Folder
Check Out	All, For this Secret, In this Folder
Copy	All, For this Secret, In this Folder
Create	All, In this Folder
Custom Audit	All, For this Secret, In this Folder
Custom Password Requirement Added to Field	All, For this Secret, In this Folder

Event	Scope
Custom Password Requirement Removed from Field	All, For this Secret, In this Folder
Delete	All, For this Secret, In this Folder
Dependency Added	All, For this Secret, In this Folder
Dependency Deleted	All, For this Secret, In this Folder
Dependency Failure	All, For this Secret, In this Folder
Edit	All, For this Secret, In this Folder
Expired Today	All, For this Secret, In this Folder
Expires in 1 Day	All, For this Secret, In this Folder
Expires in 3 Days	All, For this Secret, In this Folder
Expires in 7 Days	All, For this Secret, In this Folder
Expires in 15 Days	All, For this Secret, In this Folder
Expires in 30 Days	All, For this Secret, In this Folder
Expires in 45 Days	All, For this Secret, In this Folder
Expires in 60 Days	All, For this Secret, In this Folder
Export	All, For this Secret, In this Folder
File Save	All, For this Secret, In this Folder
Heartbeat Failure	All, For this Secret, In this Folder
Heartbeat Success	All, For this Secret, In this Folder
Hook Create	All, For this Secret, In this Folder
Hook Delete	All, For this Secret, In this Folder
Hook Edit	All, For this Secret, In this Folder
Hook Failure	All, For this Secret, In this Folder
Hook Success	All, For this Secret, In this Folder

Event	Scope
Launch	All, For this Secret, In this Folder
Password Change Maximum Attempts Reached	All, For this Secret, In this Folder
Password Copied to Clipboard	All, For this Secret, In this Folder
Password Displayed	All, For this Secret, In this Folder
Password Change	All, For this Secret, In this Folder
Secret Policy Change	All, For this Secret, In this Folder
Session Recording View	All, For this Secret, In this Folder
Undelete	All, For this Secret, In this Folder
View	All, For this Secret, In this Folder
View Secret Edit	All, For this Secret, In this Folder
Web Password Fill	All, For this Secret, In this Folder

Table: User Events

Event	Scope
Added to Group	All, For this User, For this Group
Challenge Applied	All, For this User
Challenge Cleared	All, For this User
Create	-
Disabled	All, For this User
Edit	All, For this User
Enable	All, For this User
Lockout	All, For this User
Login	All, For this User
Login Failure	All, For this User

Event	Scope
Logout	All, For this User
Owners Modified	All, For this User
Password Change	All, For this User
Removed From Group	All, For this User, For this Group
Two Factor Changed	All, For this User

Table: Other Events

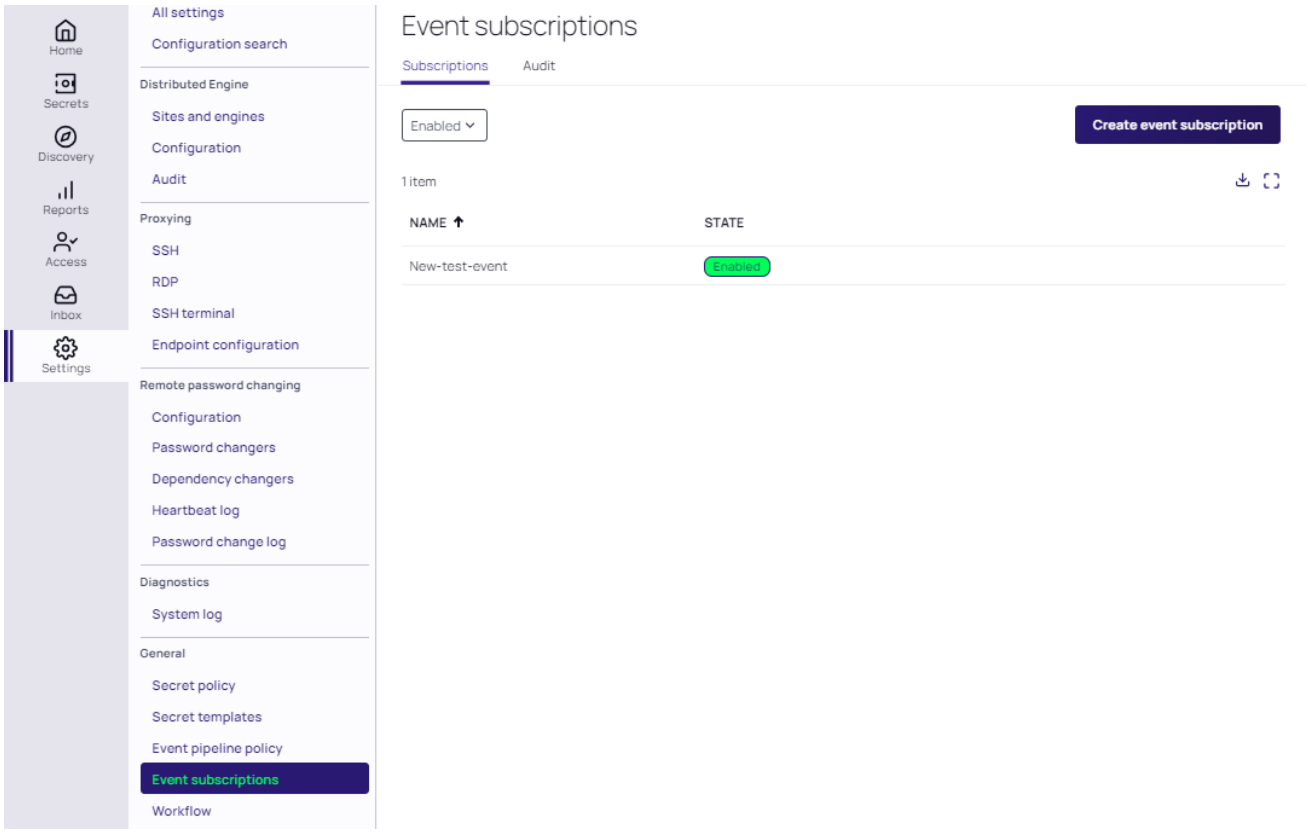
Entity	Event
Automatic Export	Download, Edit, Export, Run Export
Configuration	Edit
Dual Controls	Create, Delete, Edit
Encryption	HSM Disable, HSM Enable, Rotate Secret Keys, Rotate Secret Keys Cancel Requested, Rotate Secret Keys Failure, Rotate Secret Keys Success
Engine	Engine Activated, Create, Deactivate, Delete, Offline, Online
Export Secrets	Exported
Group	Owner Modified
Import Secrets	Imported
IP Address Range	Create, Delete, Group Assigned, Group Unassigned, Edit, User Assigned, User Unassigned
Licenses	Expires in 30 Days
Role	Assigned User or Group, Create, Edit, Role Disabled, Role Enabled, Unassigned User or Group
Role Permission	Added to Role, Removed From Role
Script - PowerShell	Create, Deactivate, Edit, Reactivate, View
Script - SQL	Create, Deactivate, Edit, Reactivate, View
Script - SSH	Create, Deactivate, Edit, Reactivate, View

Entity	Event
Secret Policy	Create, Edit
Secret Template	Create, Create Secret Access Changed, Edit, Field Encrypted, Field Exposed, Owners Modified, Copy
Privileged Behavior Analytics Configuration	Edit
Site	Engine Added, Domain Assigned to Site, Create, Disable, Edit, Enable, Engine Downloaded, Engine Offline, Engine Online, Domain Removed from Site, Engine Removed
Site Connector	Create, Credential View, Disable, Edit, Enable
Unlimited Administrator	Disable, Enable
User Audit	Expire Now

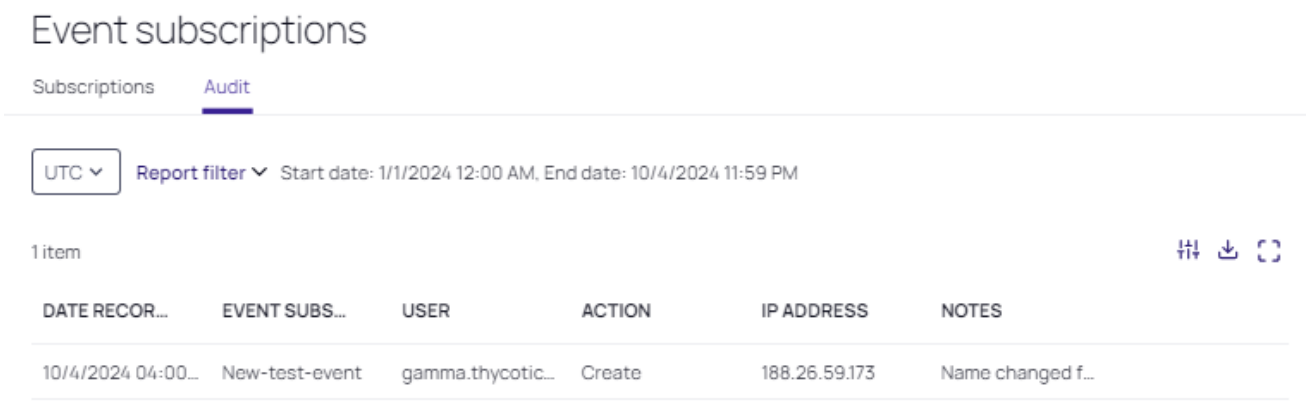
Viewing Event Subscription Logs

To view the events that have been triggered in a subscription perform the following:

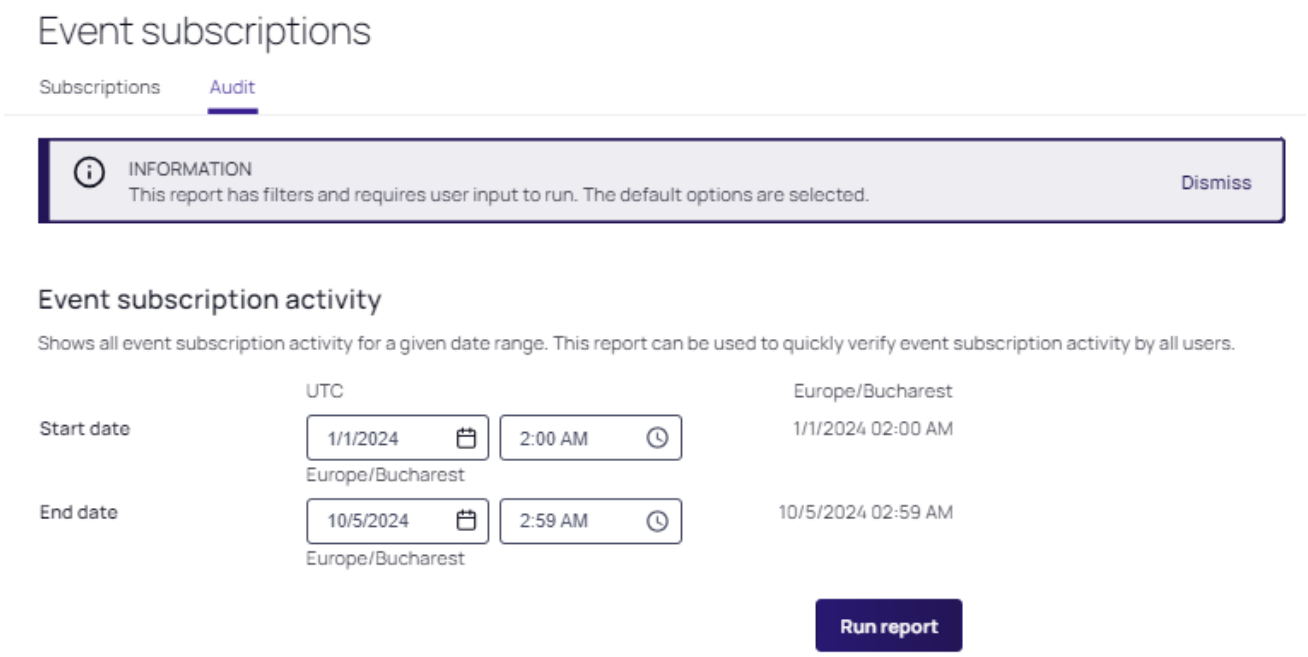
1. Navigate to **Settings > General > Event Subscriptions** it shows all current enabled event subscriptions by default.



2. Navigate to the **Audit** tab. A report is generated automatically for the event subscriptions you have enabled. The most recent events to have been triggered are typically at the top of the list.
3. To select a specific time frame, click the **Report filter** drop-down:



4. The Event subscription activity date range selectors for your current location appear here. Select the start and end dates, along with their respective times. Click **Run report** to return the corresponding log entries:



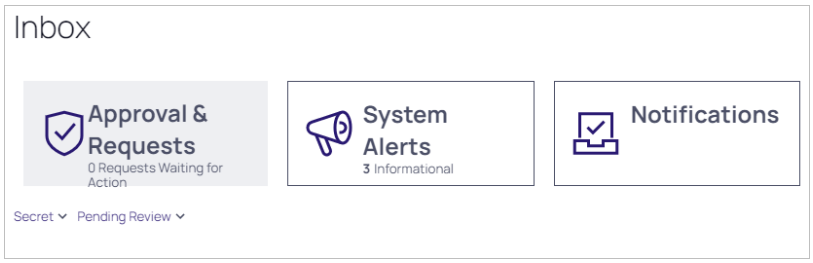
It may take a few seconds for the events to make it into the log.

Notification Inbox Overview

The notification Inbox, or simply inbox, is a centralized interface for managing notifications such as event subscription alerts, access requests and approvals, and other configuration alerts. Here are the key features and functions:

- Notifications: The Inbox displays various notifications, including event subscription alerts, access requests, and system alerts. You can configure notifications to be forwarded via email or Slack based on customizable criteria.
- Access: You can access the Inbox by clicking the Inbox button on the main menu.
- Customization: The format of email messages can be customized. Notifications can be configured to disappear from the notification center after being viewed, while system alerts and access requests remain active until resolved.
- Inbox Templates: Templates are used to format notifications. These templates can be customized and include variables that are replaced by Secret Server when sending messages. Templates can be used for both email and Slack notifications.
- Inbox Rules: Rules can be set up to trigger notifications and send them to specified users or groups via email or Slack. Rules can be created from scratch or based on existing notifications.
- Marking Alerts as Viewed: Alerts can be marked as read, which removes them from the main view but allows them to be seen again if needed.

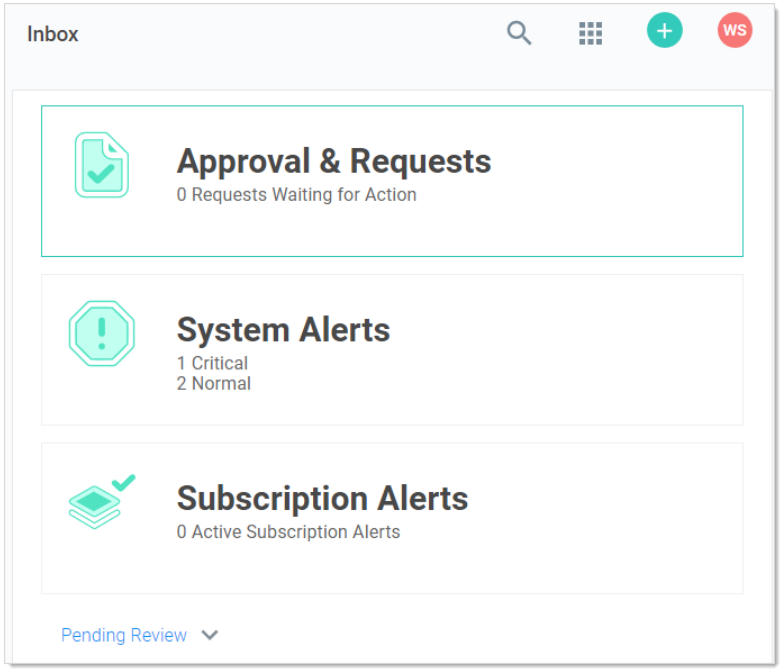
The Inbox was also called the *Alert Notification Center* in earlier Secret Server versions.



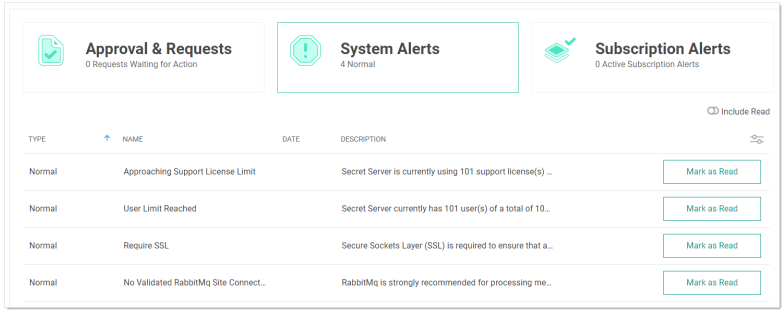
Event subscriptions disappear from the notification center after you view them. System alerts and access requests stay active until resolved.

Marking Alerts as Viewed

1. Access the alert notification center by clicking the **Inbox** button on the main menu. The Inbox appears:



2. Click the **System Alerts** button. The System Alerts page appears:



3. Click the **Mark as Read** button for the each alert you no longer wish to view. The alert disappears, but you can still see it if you click the **Include Read** toggle button.

Using Inbox Rules

Overview

An inbox rule (notification rule) triggers on notifications and sends either an email or a Slack message to a specified group of users. First, we discuss an inbox rule's components. Second, we show how to create an inbox rule from scratch and based on an existing notification.

> There are some emails types that Secret Server sends that do not go through the inbox. For example, the test email button on the email configuration page sends a plain-text email directly. There are also some diagnostic emails that are directly sent. For example, discovery can directly send a detailed log which is essentially a text dump. Inbox rules are primarily for non-admin end-user communications and event subscriptions.

> You can still send legacy emails (earlier email notifications that did not go through the inbox) if desired. Go to **Admin > Configuration > Email Tab > Enable Legacy Email** to set this up.

Inbox Rule Components

Inbox rules have the following components:

Message (Notification) Types

This is the notification message (alert) types that the rule responds to. These include:

- Dependency Failure
- Event Subscription
- Inbox Test Message
- Password Reset
- Secret Access Approved
- Secret Access Cancel Request
- Secret Access Deny Request
- Secret Access Request
- Secret Changed
- Secret Heartbeat Failed
- Secret View
- Workflow Access Approval Request
- Workflow Access Request Expired
- Workflow Access Request Incomplete

- Workflow Access Request Next Step
- Workflow Access Step Approved

Rule Conditions

Rule conditions are filters that define who receives the email or Slack message when a notification arrives. Conditions are matched with text string matching: equals, not equals, or regex. If no condition exists, the rule triggers for every message of the defined types. Condition types include:

- **ActionType:** Specific entities via text matching of the action's value or display value. Actions types vary by inbox message type. For example, "EXPORTED."

  > In the case of an event subscription notification, these are the same as the event subscription events. Other notification types have different action types.

- **Container:** Specific containers via text matching of the container name. Containers include folders (secret containers) and roles (quasi user containers).
- **Details:** Specific text string in the details section. The Details type serves as a summary of the message for display in the inbox. Sometimes it contains information that is from other condition types. For example: "App Settings Exported - SECRETSERVERSETTINGS - EXPORTED."
- **EntityType:** Specific entities via text matching of the entity name or value. Entities are what is having the action done to it, for example, "SECRETSERVERSETTINGS." These are the same as the event subscription entities.
- **EventDetails:** Specific text string in the Event Details section. For example: "Application Settings," "Launcher Settings," and "Protocol Handler Settings."
- **ItemName:** The source item. Specific items via text matching of the item's value or display value. The "item" varies by message. For an event subscription secret action it contains the secret name. For an event subscription folder action, it contains the folder name.
- **SubscriptionName:** The source event subscription. Specific subscriptions via text matching of the subscription's value or display value.
- **User:** The user that created the rule. Specific users via text matching of the username's value or display value.
- **Rule Subscribers:** What users receive the action result (an email or Slack message).
- **Rule Actions:** What actions the rule performs:
  - Send to an email address using a specific HTML template, which the user defines.
  - Send to Slack using a specific markdown template, which the user defines.

Predefined System Rules

Secret Server ships with several predefined system rules in inbox templates. These rules can only be disabled or enabled. However, you can copy a system template to your own custom template and edit that. This allows us to upgrade system rules without interfering with your customizations. The predefined system rules are:

- Dependency Failure
- Inbox Test Message

- Password Reset
- Secret Access Deny Request
- Secret Access Request
- Secret Changed
- Secret Heartbeat Failed
- Workflow Access Approval Request

Example Rule Diagram

> This example diagram is specific to event subscription notifications. There are many other types that differ slightly, especially in content.

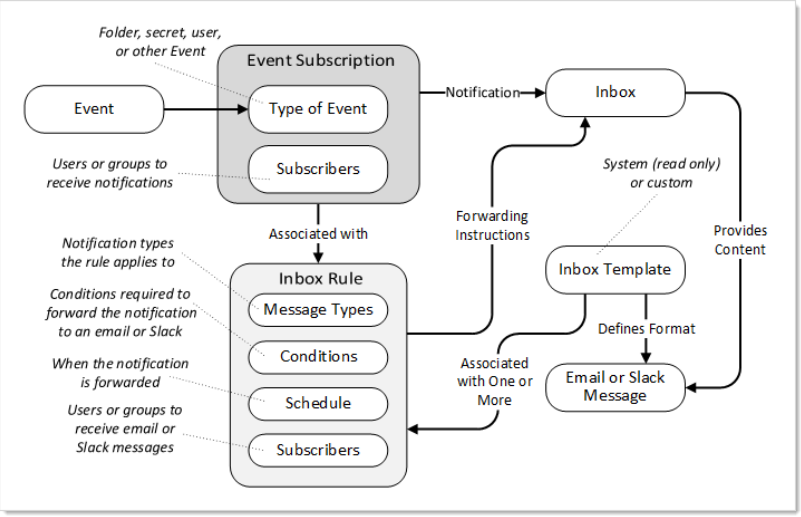
In the following diagram:

1. A secret triggers an event matching an event subscription.
2. Secret Server notifies the users and groups on the event subscription subscribers list. The notifications appear in their inboxes.
3. The inbox rule associated with the event subscription evaluates the conditions for forwarding the notification via email or Slack.
4. The inbox rule checks its schedule to determine when to forward the message.
5. The inbox rule checks its subscribers list to determine who to forward the message to.

   > Remember, arriving notifications may have already been filtered by whose inbox gets the notification in the first place. Thus, the inbox rule could be set to sent to everybody but only those who receive the notification in their inbox will receive the email or Slack message.

6. The inbox rule references the message type's inbox template to format and populate the message's variables.
7. When the scheduled time arrives, which can be immediately, Secret Server sends the messages to the subscribers.

Figure: Event Subscription Using an Inbox Rule to Forward a Notification via the Inbox



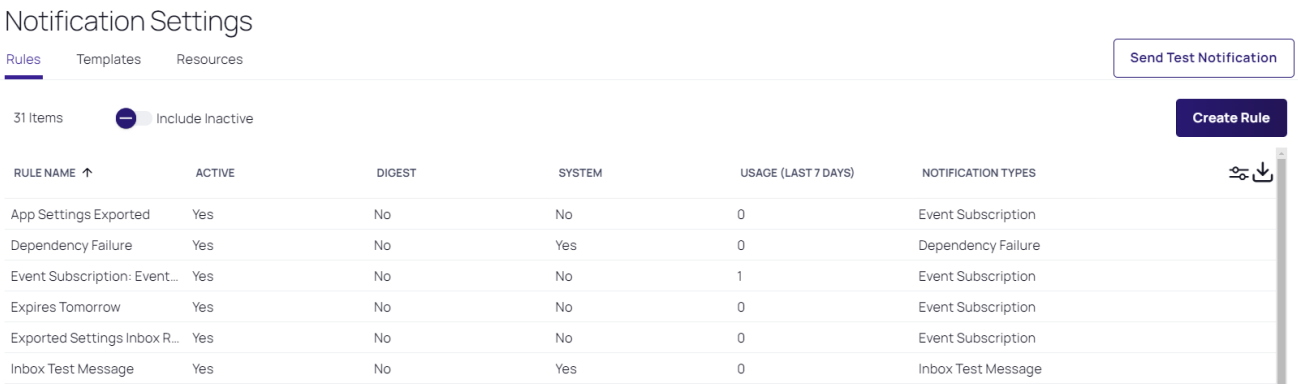
Procedures

Creating a Rule from Scratch

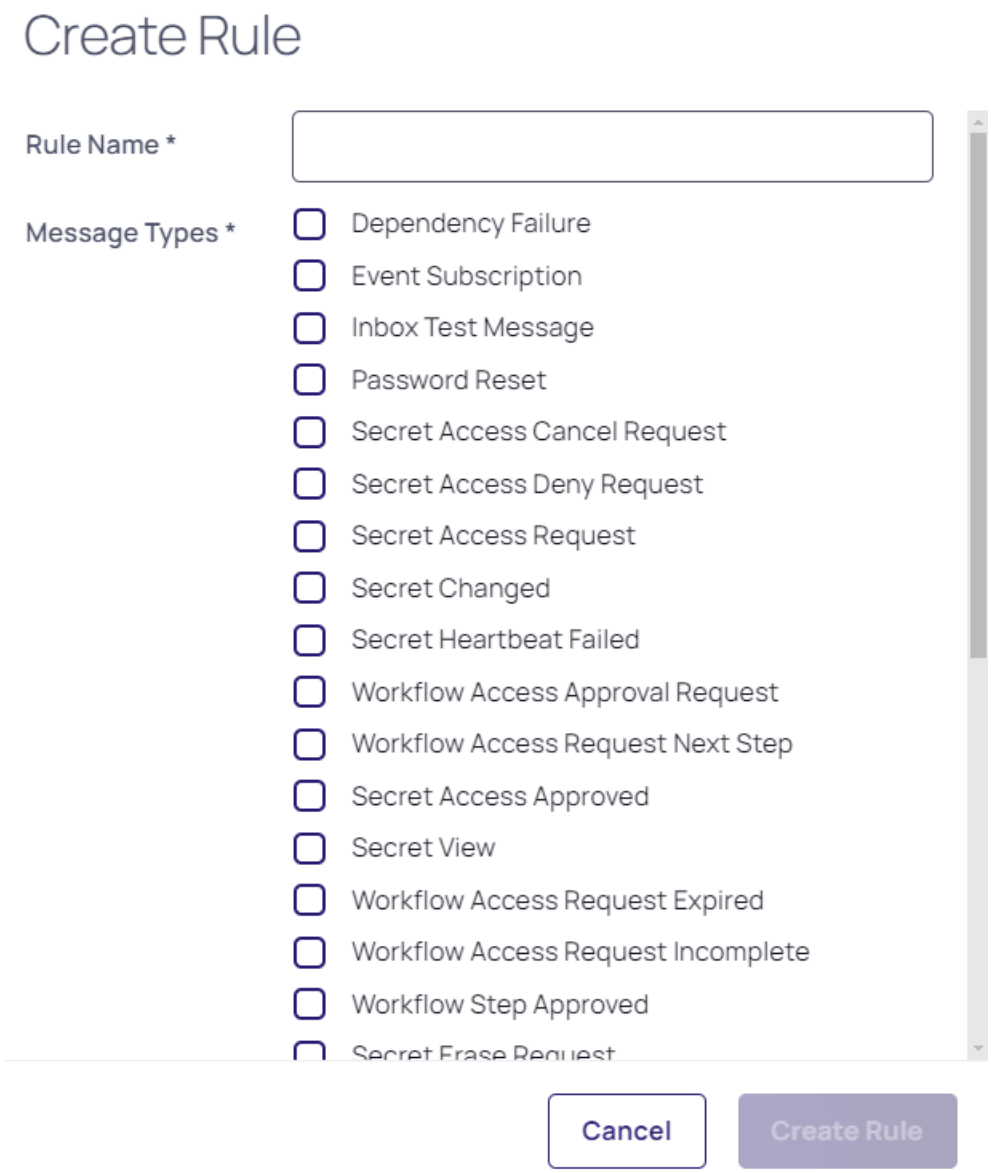
Let us say we want to be notified when anybody tries to edit the permissions on the "No-Go Secrets" folder. This is an event, so the notification type will be an event subscription.

Task 1: Create the Inbox Rule

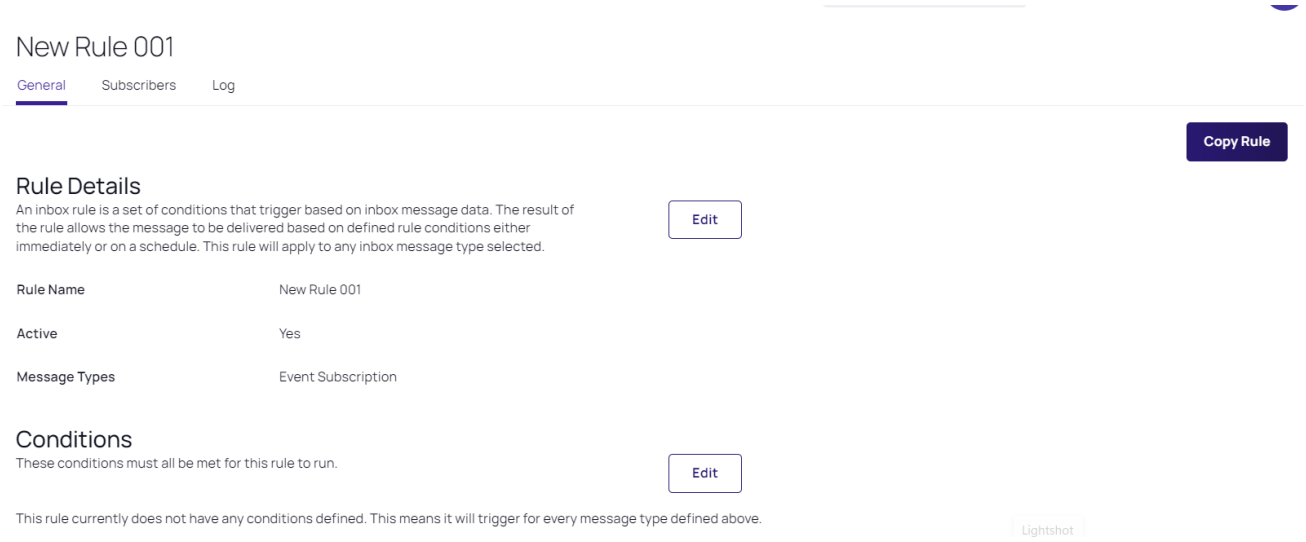
1. Select **Admin** > **Notification Rules and Templates** The Notification Settings page appears:



2. Click the **Create Rule** button. The Create Rule popup appears:



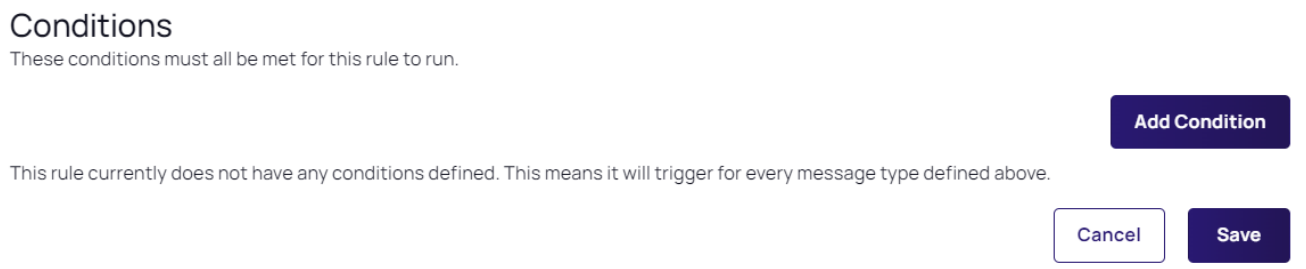
3. Type the new rule's name in the **Rule Name** text box.
4. Click to select the **Message Types** check box for the message types you wish to apply the new rule to. (**Event Subscription** for this case).
5. Ensure the **Active** check box is selected.
6. Select the desired radio button for **Action** (**Email** or **Slack**).
7. If necessary, scroll down to the bottom of the popup.
8. Click the **Template** dropdown list to select the desired inbox template to associate the rule with.
9. Click the **Create Rule** button. The configuration page for the new rule appears:



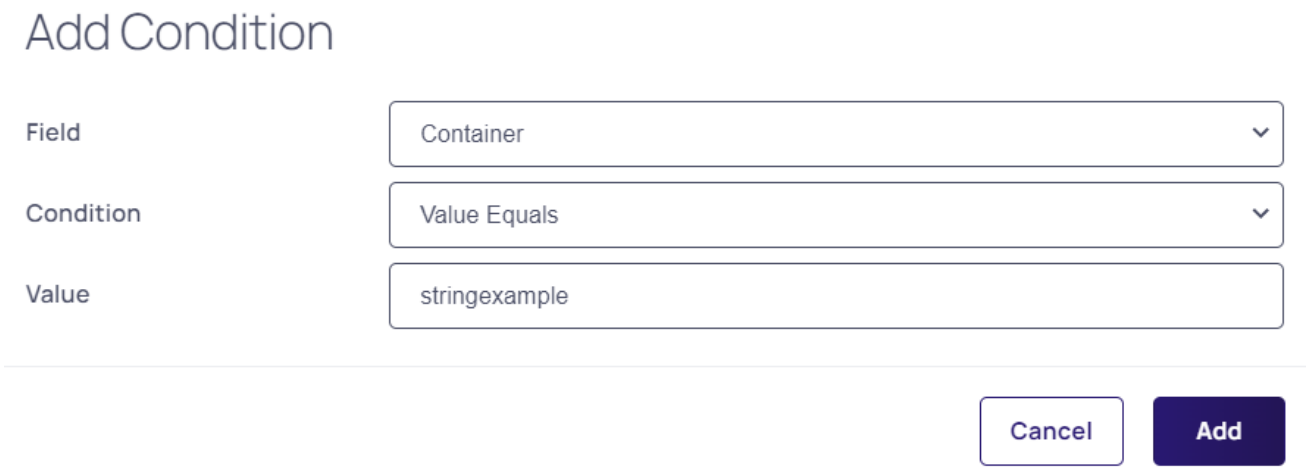
Task 2: Add Rule Conditions

1. Click the **Edit** button in the **Conditions** section.

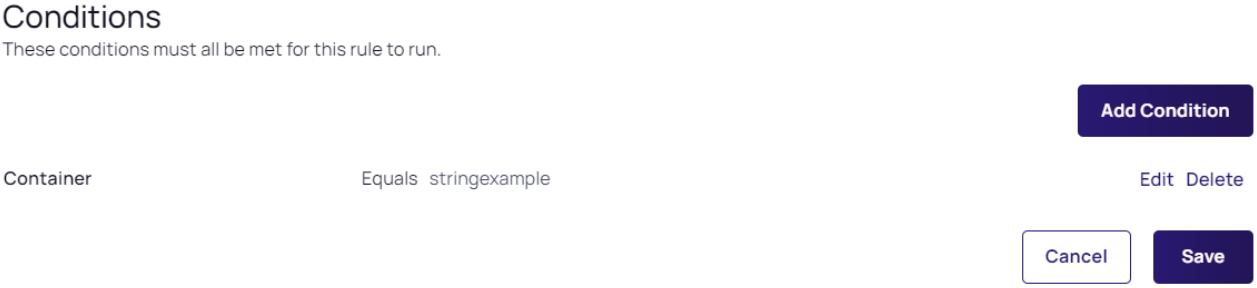
   If you do not add any conditions, the rule will apply to all messages of the types you chose. New buttons appear:



2. Click the **Add Condition** button. The Add Rule Condition popup appears.
3. Click the **Field** dropdown list to select the type of rule condition. For this instruction, we chose Container (a folder). A Condition dropdown list appears:
4. Click the **Condition** dropdown list to select how the value (added next) is compared to the message. Choose "Value RegEx" if you want to create a regular expression to further refine the condition. We chose Value Equals. The Value text box appears. Type the string you want to test for in the **Value** text box.

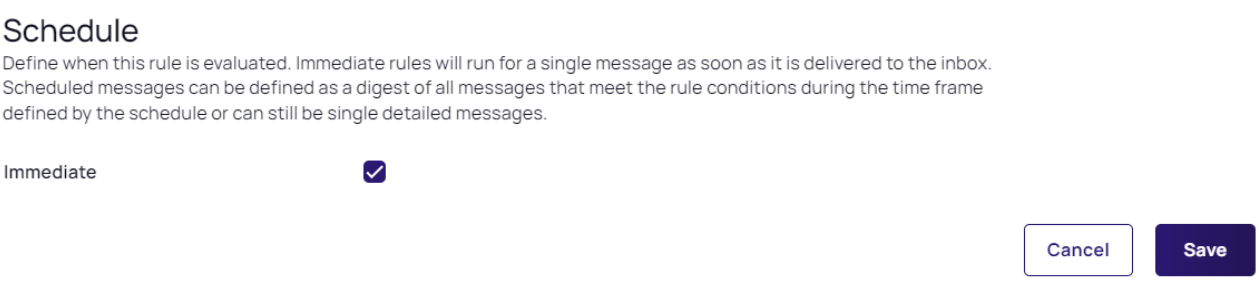


5. Click the **Add** button. The new condition appears:



It tells us the rule is triggered if there is an associated event.

6. Click the **Save** button.
7. Click the **Edit** button in the **Schedule** section. The section becomes editable:

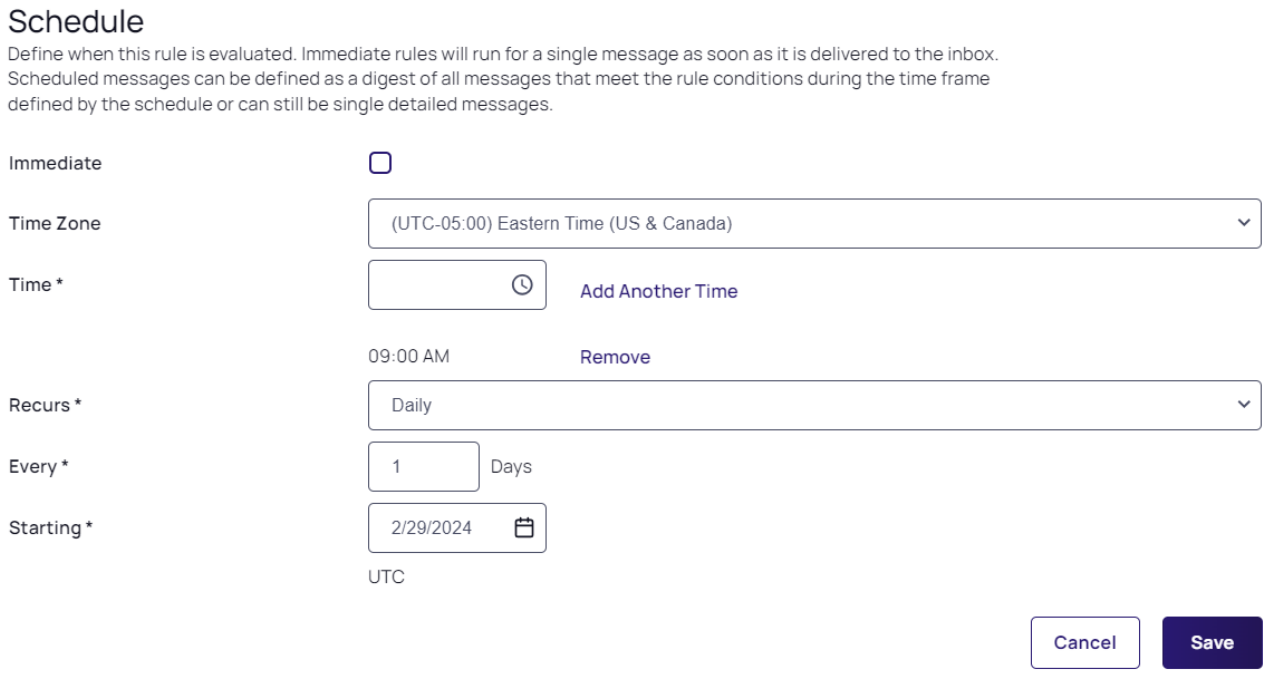


> Any schedule choice other than Immediate produces a digest (summary) of notifications. Users receiving a digest can click on individual entries to see the notification. The following instructions show how to set up a digest.

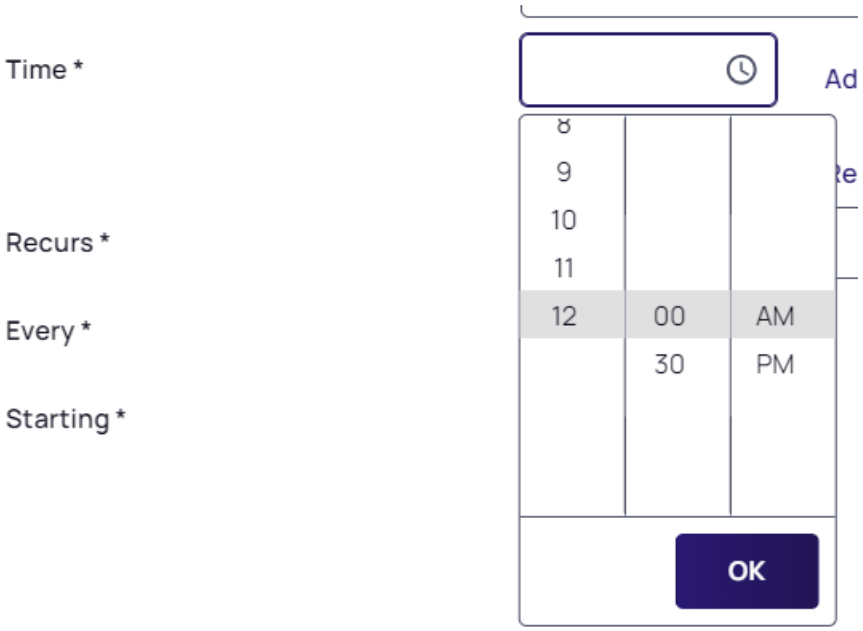
8. If you want to send one email per notification, ensure the Immediate check box is selected, and skip to Task 4.

Task 3: Set up an Email Digest

1. If you do not want the notifications sent immediately, click to deselect the Immediate check box. A time setting section appears:



2. Click the **Timezone** dropdown list to select a time zone for the scheduled notification.
3. Click the clock icon to select a time to add. A time setting table appears:

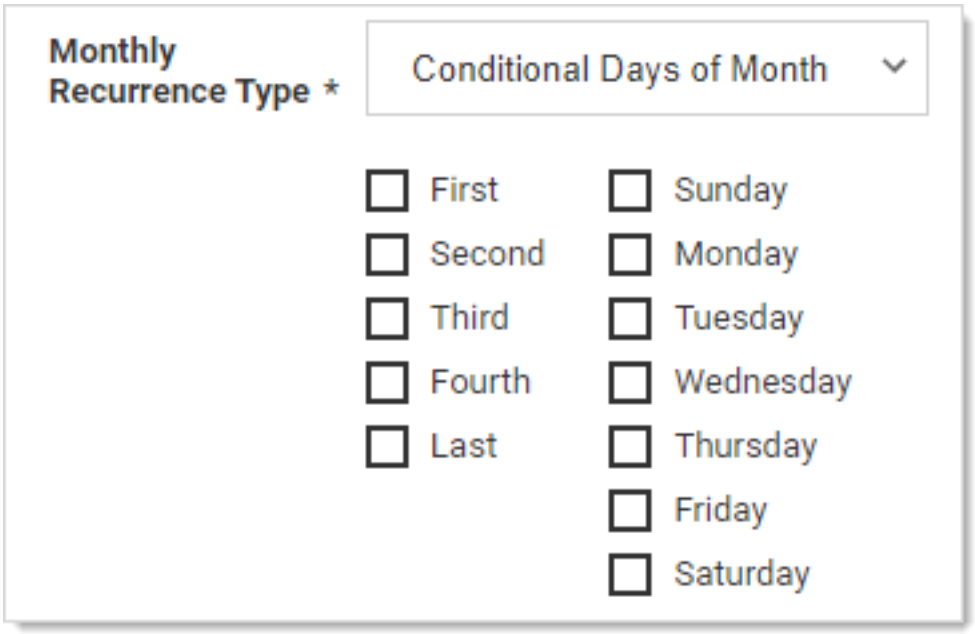


4. Hover the mouse over the one of the columns for a scrollbar appears. Click or drag to select the hour, half hour, or AM/PM.

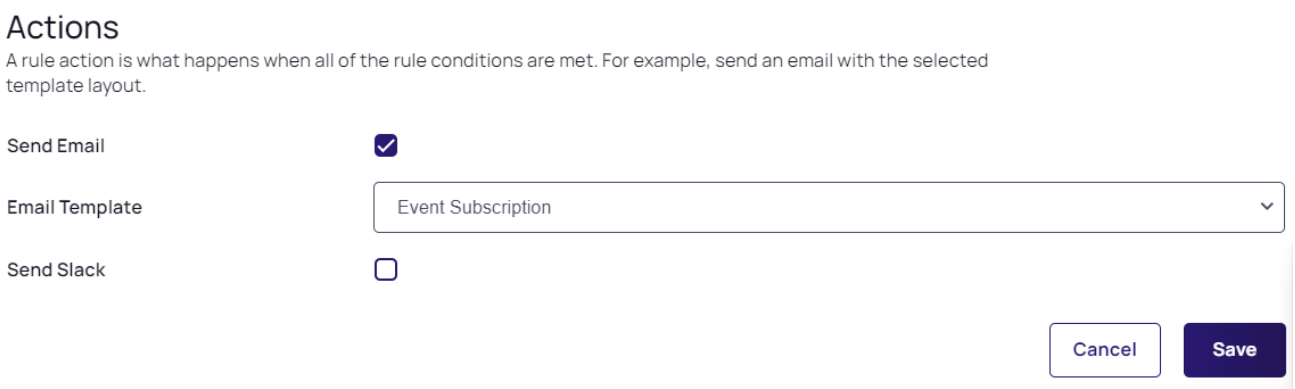
> You also can simply type the time prior to clicking the clock icon in the format hh:mm AM/PM. If you choose minutes other than 00 or 30, it will be converted to the nearest 00 or 30 when you input it.

5. Click the **OK** button. The time appears in the text box.
6. Click the **Add Another Time** link to commit the time.
7. Add more times as desired.
8. Click the **Recurs** dropdown list to choose the frequency. The Every section will change, depending on what you chose:
   - Daily: Type the number of days that pass between notifications in the **Days** text box.
   - Weekly: Type the number of weeks that pass between notifications in the **Weeks** text box. Click to select the days of the week check box to select which days to send the notifications.
   - Monthly: Click the **Monthly Recurrence Type** dropdown list to select either specific or conditional days of the month. The former provides a calendar to choose which days. The latter provides check boxes for selecting relative days where you choose the day of the week and the position in the month, for example,

Last and Friday.



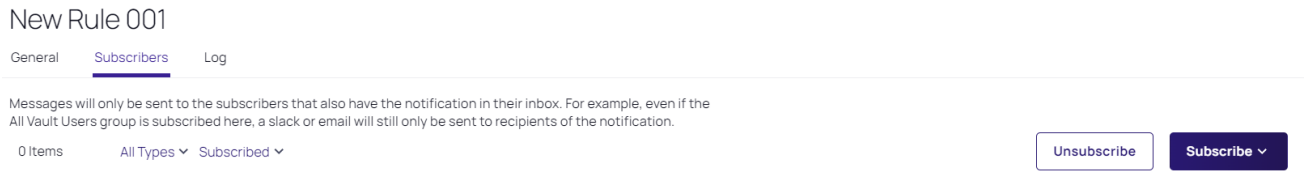
9. Click the **Starting** calendar icon to select the date when you want to start the notification schedule.
10. Click the **Save** button to commit your choices.
11. Scroll down to the **Actions** section.
12. Click the **Edit** link next to **Actions**. The section becomes editable:



13. Click to select either the **Send Email** or **Send Slack** check box (or both).
14. If you chose to send email, click the **Email Template** dropdown list to choose the email format (inbox template). There are several standard ones, and you can customize your own. For this instruction, we choose **Standard Email** if did not create a digest and **Standard Email Digest** if we did. See "Using Inbox Templates" on page 334 for details.
15. Click the **Save** button

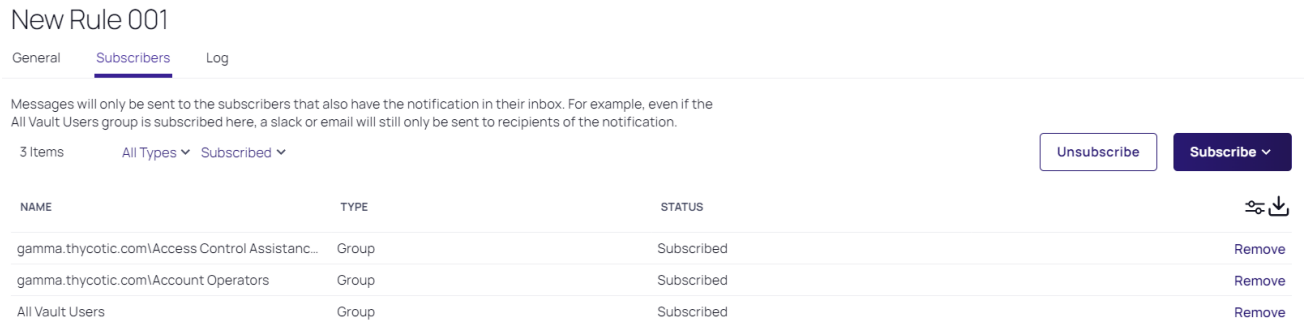
Task 4: Add Subscribers to the Email or Slack Message

1. Click the **Subscribers** tab:



2. Click the Subscribe button and select **Users**, **Groups**, or **External Emails**. Users provides a list of users for you to select from. Groups provides a list of groups for selection. External Emails provides a text box to enter a specific email to somebody without a Secret Server account. For this instruction, we added one of each:

> External emails get sent without regard to whether somebody has a notification in their inbox because external "users" do not have an inbox.

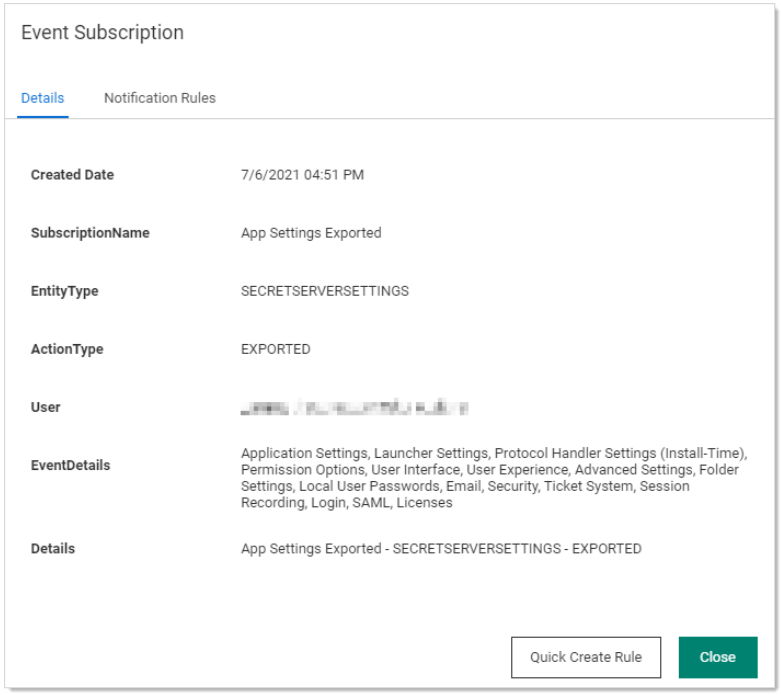


3. You can return to this page at a later date to edit this list. You can also unsubscribe users that are members of a subscribed group without actually removing them from the list for ease of subscribing them later.

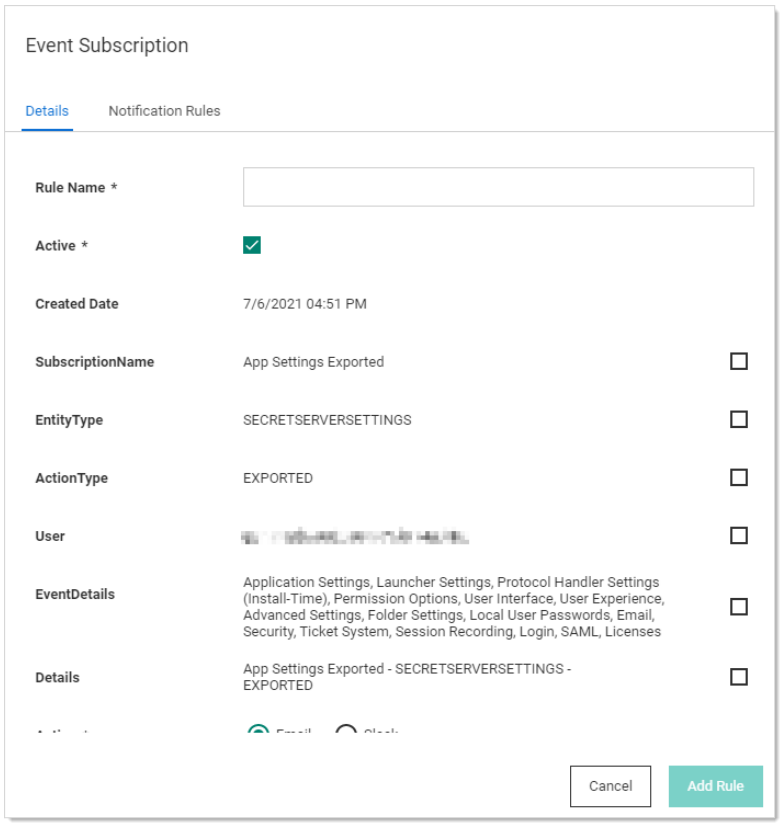
Creating an Inbox Rule from a Notification

If a inbox notification is what you want to forward to an email or Slack message from here on out, we provide a shortcut feature that allows you to quickly build an inbox rule from the notification. To create the inbox rule:

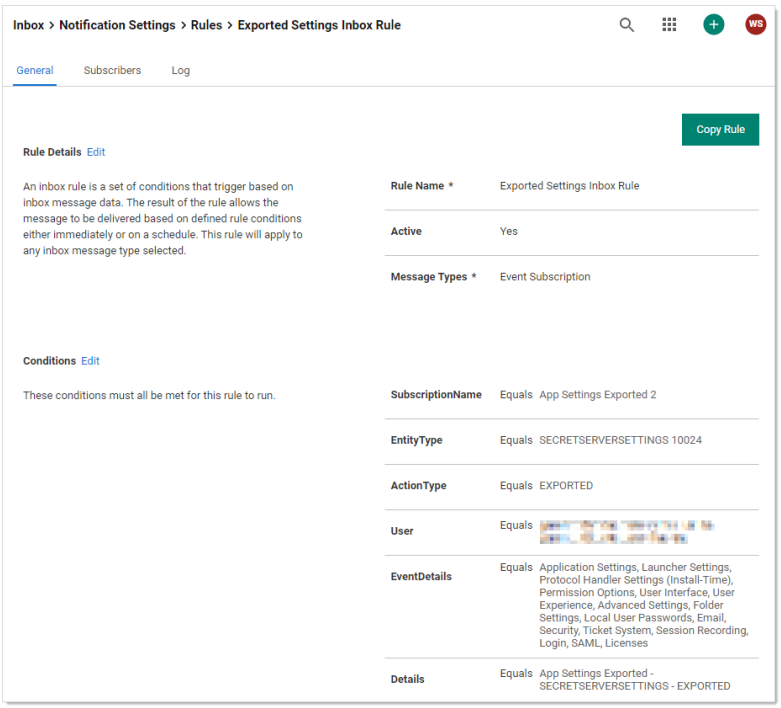
1. Click a notification in the inbox. A details popup appears. For instance:



2. If you desire an inbox rule to react to that sort of message, click the **Quick Create Rule** button. A very similar, editable page appears:



3. Type the new rule's name in the **Rule Name** text box.
4. Click to select the check box for each rule component you want to include.
5. Click to select the **Action** selection button for the type of notification.
6. If necessary, scroll down to the bottom of the popup.
7. Click the **Template** dropdown list to select the desired inbox template to associate the rule with.
8. Click the **Add Rule** button. The configuration page for the new rule appears:



9. Edit the rule as desired. See Creating a Rule from Scratch.

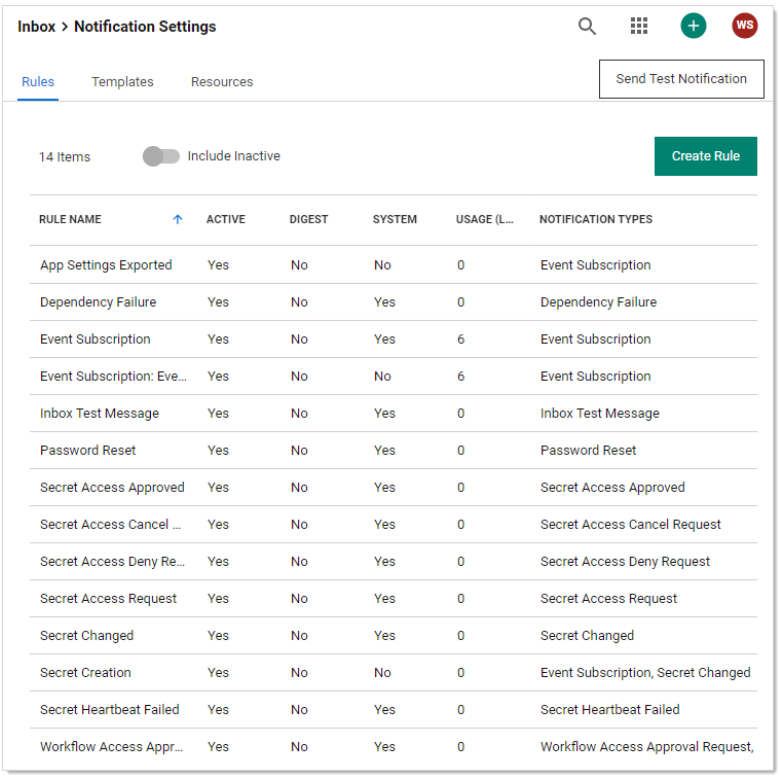
Using Inbox Templates

Overview

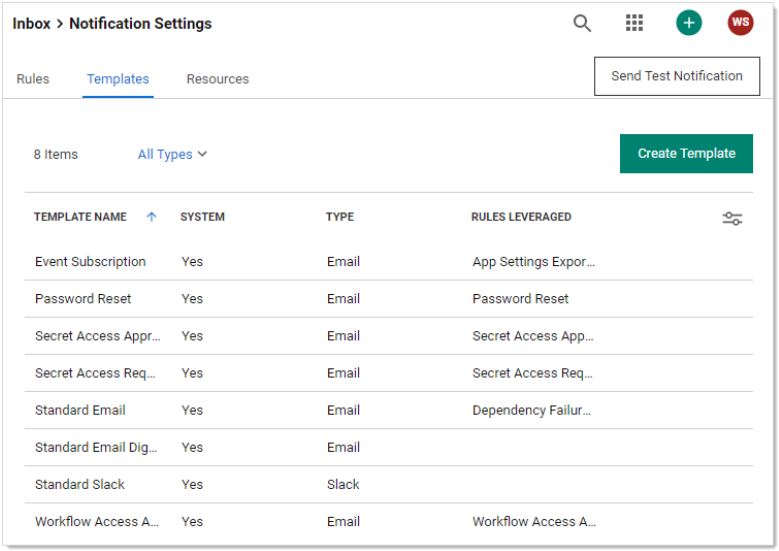
> Inbox templates are also used by "Event Pipelines" on page 285. When used for event pipelines, the templates can use event pipeline tokens in addition to the standard inbox template tokens in the message subject and body. Event pipeline messages are sent directly to the user's inbox based on the email addresses generated by the send email event pipeline task, not subscribers from any inbox rule.

First, let us open a system inbox template to look at:

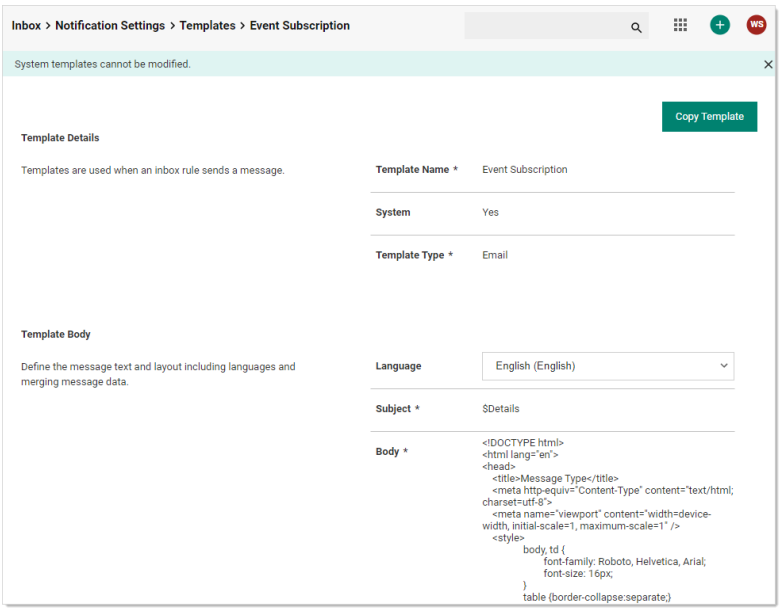
1. Go to **Admin > All**.
2. If necessary, click the view link to switch to **Alphabetized View**.
3. Click the **Notification Rules and Templates** link. The Notification Settings page appears:



4. Click the **Templates** tab:



5. Note that most, if not all, of the inbox templates are system templates. These are **read-only** templates that you can clone to create custom templates. That is, system templates are templates for your templates. Most of the templates are email templates, and one is a Slack template.
6. Click the Template Name for the Event Subscription template. The template's page appears:



7. Note that each template has:
   - A details section that contains the name, a system template flag, and a template type (email or Slack).
   - A body section that defines the subject, language used, and the canned text for the message. The message contains variables that are drawn from the alert or event. The body is read only in system templates and is editable in custom templates cloned (copied) from system templates.
   - Zero or more associated inbox rules. These are the inbox rule types that use this inbox template (message type). Rules define filters for the alerts or events (what characteristics trigger the rule) and who gets externally notified via email or Slack (the subscribed users or groups). The following table lists the system templates and their associated inbox rules that use them.
   - Zero or more resources. These are items, such as images, that go along with any email based on the template.

Table: System Inbox Rules by Inbox Template

System Inbox Template	Type	Associated Inbox Rules
Event Subscription	Email	App Settings Exported Event Subscription
Password Reset	Email	Password Reset
Secret Access Approved	Email	Secret Access Approved
Secret Access Request	Email	Secret Access Request

System Inbox Template	Type	Associated Inbox Rules
Standard Email	Email	Dependency Failure Inbox Test Message Secret Access Cancel Request Secret Access Deny Request Secret Changed Secret Heartbeat Failed
Standard Email Digest	Email	
Standard Slack	Slack	
Workflow Access Approval Request	Email	Workflow Access Approval Request

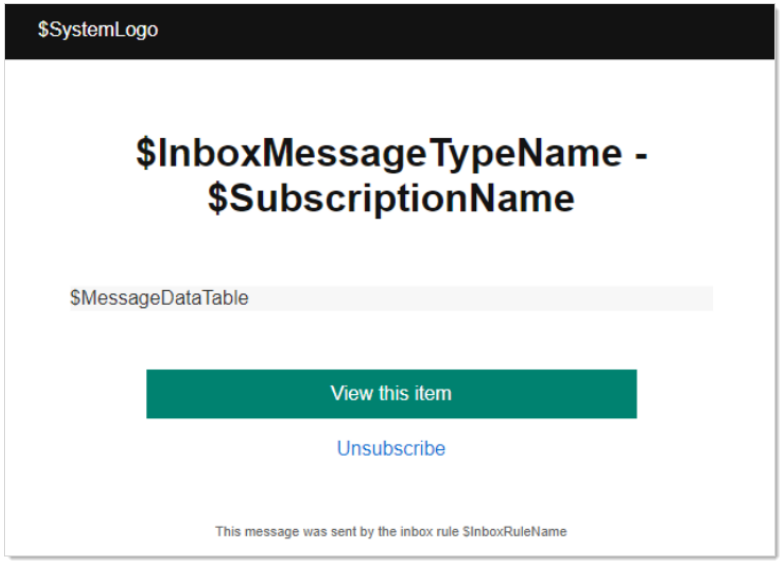
8. Note the Body section is HTML for emails and Slack template text. For the Event Subscription template it looks like this:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Message Type</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1" />
    <style>
        body, td {
            font-family: Roboto, Helvetica, Arial;
            font-size: 16px;
        }
        table {border-collapse:separate;}
        a, a:link, a:visited {text-decoration: none; color: #1071D4;}
        a:hover {text-decoration: underline;}
        h2,h2 a,h2 a:visited,h3,h3 a,h3 a:visited,h4,h5,h6,.t_cht {color:#000 !important;}
        .ExternalClass p, .ExternalClass span, .ExternalClass font, .ExternalClass td
{line-height: 100%;}
        .ExternalClass {width: 100%;}
        h1 { color: #121212; font-family: Roboto, Helvetica, Arial;font-style:
normal;font-weight: bold;font-size: 32px; }
    </style>
</head>
<body style="background-color: #F7F7F7;">
  <table width="100%" border="0" cellspacing="0" cellpadding="0"><tr><td align="center">
  <table cellspacing="0" cellpadding="0" border="0" width="100%" style="max-width: 640px;
">
    <tr>
        <td style="background-color: #121212;width: 80%; height: 48px; color: #ffffff;
padding-left: 32px;">
            $SystemLogo
        </td>
```

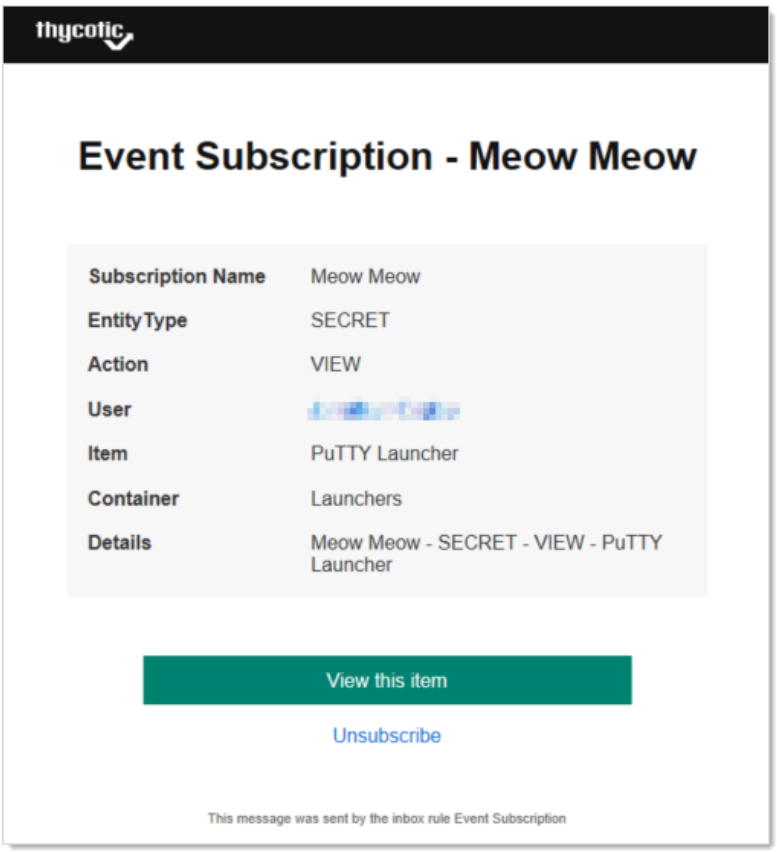
```
    </tr>
    <tr>
        <td style="background-color:#ffffff">
            <table cellspacing="16" width="100%">
                <tr>
                    <td width="24"> </td>
                    <td align="center" style="padding-top: 42px; color: #121212; font-
family: Roboto, Helvetica, Arial;font-style: normal;font-weight: bold;font-size:
32px;text-align: center;">
                        $InboxMessageTypeName - $SubscriptionName
                    </td>
                    <td width="24"> </td>
                </tr>
                <tr>
                    <td> </td>
                    <td style="font-family: Roboto, Helvetica, Arial; font-weight: normal;
font-size: 16px; color: #323232;">
                         
                    </td>
                    <td> </td>
                </tr>
                <tr>
                    <td> </td>
                    <td style="background-color: #F7F7F7; font-family: Roboto, Helvetica,
Arial; font-weight: normal; font-size: 16px; color: #323232; padding: 0px">
                        $MessageDataTable
                    </td>
                    <td> </td>
                </tr>
                <tr>
                    <td> </td>
                    <td style="font-family: Roboto, Helvetica, Arial; font-weight: normal;
font-size: 16px; color: #323232; padding: 16px" align="center">
                        <p>
                            <a
href="$ApplicationUrl/app/#/inbox/view/notifications?messageId=$MessageId" style="text-
decoration: none; display: inline-block; background-color: #008270; width: 400px; height:
40px; line-height: 40px; color: #ffffff; text-align: center">
                                View this item
                            </a>
                        </p>
                        <p>
                            <a
href=
"$ApplicationUrl/app/#/inbox/view/notifications?messageId=$MessageId&unsubscribe=true">
                                Unsubscribe
                            </a>
                        </p>
                    </td>
                    <td> </td>
                </tr>
                <tr>
                    <td> </td>
```

```
                    <td style="font-family: Roboto, Helvetica, Arial; font-weight: normal;
font-size: 11px; color: #646464" align="center">
                        This message was sent by the inbox rule $InboxRuleName
                    </td>
                    <td> </td>
                </tr>
            </table>
        </td>
    </table>
    </td></tr></table>
</body>
</html>
```

9. Rendered, the body looks like this:



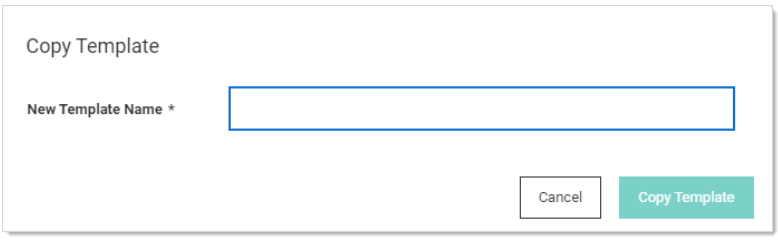
Note the variables starting with $ that are in the message. These are replaced by Secret Server when it sends the message. For example:



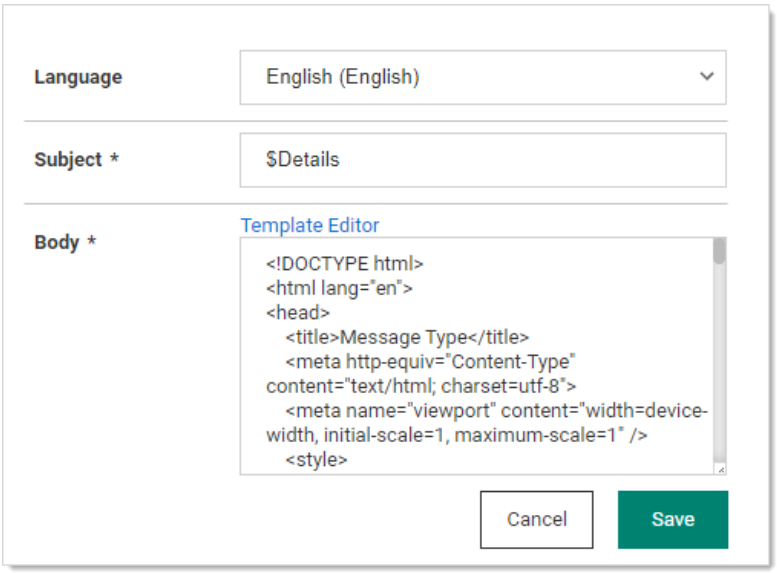
10. The variables here include:
    - `$InboxMessageTypeName` was replaced by the inbox template type.
    - `$InboxRuleName` was replaced by the inbox rule that sent the message. In this case, it is the same name as the inbox template type–Event Subscription.
    - `$MessageDataTable` was replaced by an entire table that summarized the message.
    - `$SubscriptionName` was replaced by the event subscription name.
    - `$SystemLogo` was replaced by the image resource containing the Delinea logo.

    For a complete list of variables for the template, go to the template editor (see below).

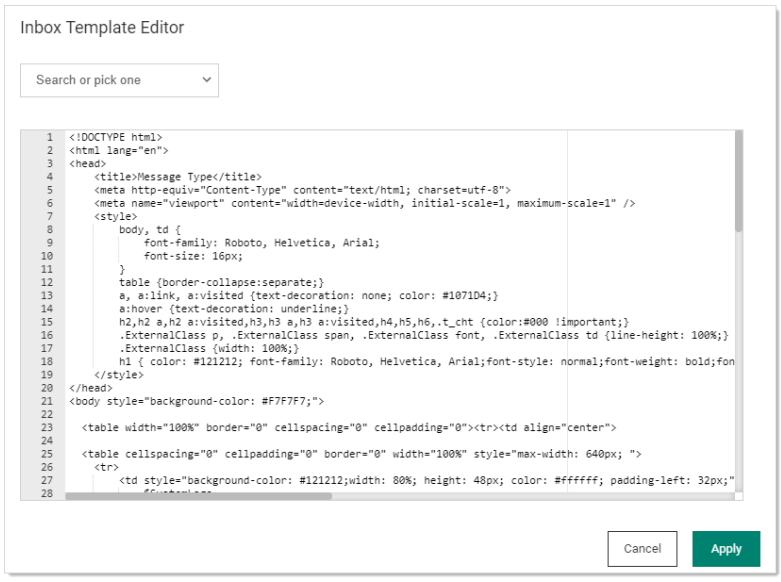
11. Nearly the entire template HTML is customizable once you make a customized clone of the system template. To clone the template click the **Copy Template** button at the top. The Copy Template popup appears:



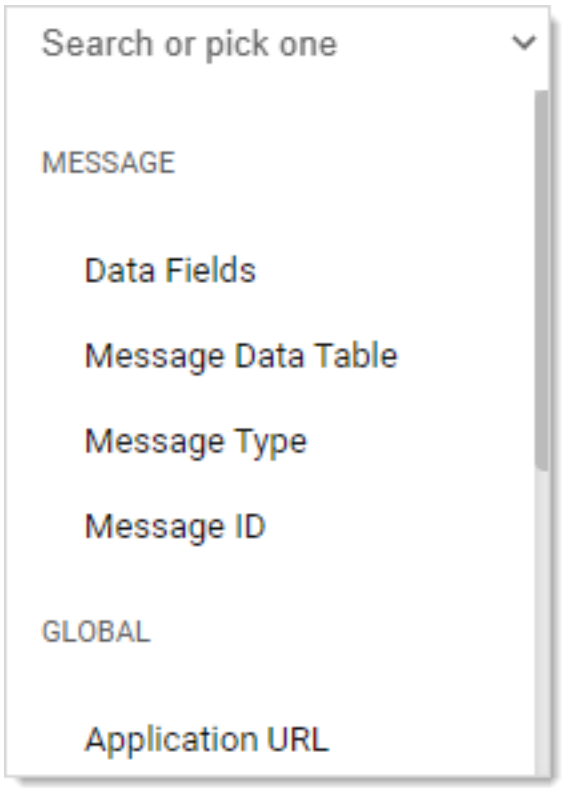
12. Type the name of the new template in the **New Template Name** text box.
13. Click the **Copy Template** button. The template page reappears, but this time it is editable and named differently.
14. Click the **Edit** link next to **Template Body**. The section becomes editable:



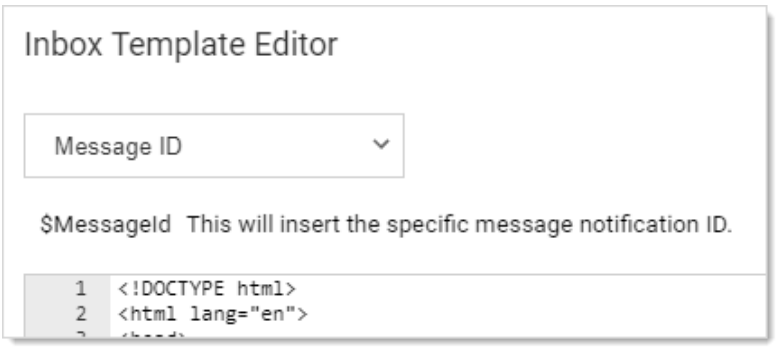
15. You can directly edit the HTML, but if you intend to add variables for Secret Server to fill in, click the **Template Editor** link. The Inbox Template Editor popup appears:



16. In addition to directly editing the HTML, you can insert variables by clicking the **Search or pick one** dropdown list:



17. You have three categories of variables: message, global, and digest.
18. Click the desired variable. The dropdown changes to show your choice:



19. The variable appears immediately below the dropdown list, as well as a description of the variable.
20. Copy or type the variable in the desired location in the HTML.
21. Edit and insert more variables as desired.
22. Click the **Apply** button. The popup disappears.
23. Click the **Save** button.

Logging Overview

Secret Server provides robust logging capabilities to ensure comprehensive tracking and auditing of all activities within the system. Here is an overview of the logging features in Secret Server:

Key Logging Features

- Syslog and CEF Logging:
  - Syslog: A standard protocol used for sending and receiving log and event messages between network devices, servers, and applications.
  - CEF (Common Event Format): A standardized log format for capturing and transmitting security-related events across various systems and devices.
- Secret Server can send log messages to an external syslog server using protocols like UDP, TCP, and Secure TCP (TLS).
- Configuration options include setting the syslog server, port, protocol, time zone, and date-time format.
- Secure TCP is recommended for sensitive log data to ensure encrypted transmission.

Secret Server Log List

- Distributed Engine Log: Logs distributed engine activity.
- Protocol Handler Log: Logs protocol handler activity, including RDP and SSH sessions.
- SS log: The main system log that reports when roles start and stop and any activity occurring on the site.
- SS-BSSR log: Logs jobs triggered by the background scheduler.

- SS-BWSR log: Logs work triggered by the background scheduler and legacy monitors, including heartbeat, password changing, and discovery.
- SS-EWSR log: Logs responses from distributed engines.
- SS-MMSR log: Logs internal site connector activity when RabbitMQ is not used.
- SS-SRWSR log: Logs session recordings.

Secret Server Log List

This topic only applies to **Secret Server On-Premises**.

Below is a collection of log lists that can be used with Secret Server.

Secret Server Logs

SS log

The Secret Server system log is a top-level IIS log that reports when roles start and stop, along with any activity occurring on the site, as well as any legacy monitors.

Location: `C:\inetpub\wwwroot\SecretServer\log`

Please refer to "Setting the Logging Levels" on page 378 for more information. See "Enabling Debug Mode in System Logs " on page 271for more information.

SS-BSSR log

The **background scheduler server role log** is responsible for jobs that fire upon a trigger. Currently, we have some monitors that schedule work from the website but will transition to trigger jobs in the scheduler.

Location: `C:\inetpub\wwwroot\SecretServer\log`

SS-BWSR log

The **background worker server role log** is responsible for logging work triggered by the background scheduler and legacy monitors. Work includes heartbeats, password changing, discovery, event pipelines, and more.

Location: `C:\inetpub\wwwroot\SecretServer\log`

SS-EWSR log

The **engine worker server role log** is responsible for processing all responses from distributed engines, such as discovery and heartbeats.

Location: `C:\inetpub\wwwroot\SecretServer\log`

SS-MMSR log

The **MemoryMq server role log** records internal site connector activity when RabbitMQ is not installed or used.

Location: `C:\inetpub\wwwroot\SecretServer\log`

SS-SRWSR log

The **session recording worker server role log** is responsible for processing session recordings from Secret Server.

Location: `C:\inetpub\wwwroot\SecretServer\log`

Protocol Handler Log

SS-RDPWin log

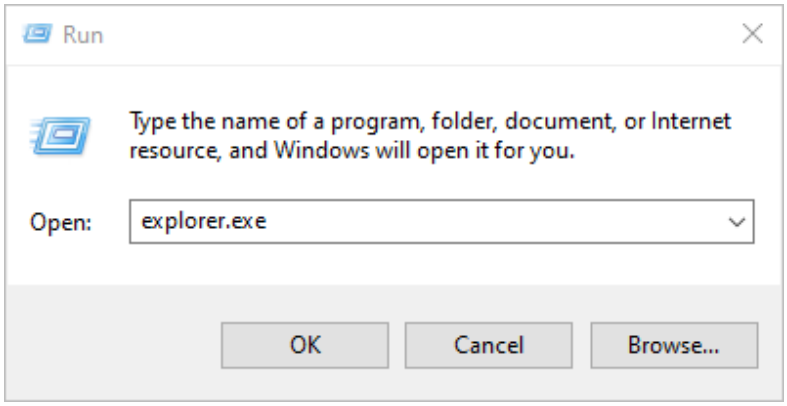
The **Protocol Handler log** records protocol handler activity among other features.

Location: stored per user on the workstation, search for: `%AppData%\Thycotic\log\SS-RDPWin.log`.

> Despite the name, the log covers more than just RDP.

To access the log:

1. Type `Run` in the Windows start menu search text box to launch the **Run Command** application. The Run popup appears:



2. Type `%AppData%\Thycotic\log` into the **Open** text box.
3. Click the **OK** button. The folder containing the file appears.

Enabling Debug Logging for the Protocol Handler Log

This section explains how to enable verbose debug logging for the Protocol Handler:

1. Navigate to `C:\Program Files\Thycotic Software Ltd\Secret Server Protocol Handler`.
2. Open the `log4net-rdp.config` configuration file in a text editor as an administrator:

```
<log4net>
    <root>
        <!--<level value="DEBUG" />-->
        <!--<level value="VERBOSE" />-->
        <!--<level value="OFF" />-->
```

```
            <level value="INFO" />
            <appender-ref ref="Thycotic.LogFileAppender" />
        </root>
        <appender
    name="Thycotic.LogFileAppender" type="log4net.Appender.RollingFileAppender">
            <!--<file value="C:\LogFiles\Thycotic\SS-RDPWin.log" />-->
            <file value="${AppData}\Thycotic\log\SS-RDPWin.log" />
            <rollingStyle value="Size" />
            <maxSizeRollBackups value="34" />
            <maximumFileSize value="10MB" />
            <lockingModel type="log4net.Appender.FileAppender+MinimalLock" />
            <layout type="log4net.Layout.PatternLayout">
                <conversionPattern value="%utcdate [CID:%property{Correlation}] [C:%property
    {Context}] [TID:%thread] %-5level %logger - %message%newline" />
            </layout>
        </appender>
    </log4net>
```

3. Comment out `<!--<level value="INFO" />`.
4. Remove the comment out of `<level value="DEBUG" />`.
5. Recreate the original issue a couple of times with DEBUG enabled.
6. Navigate to `~\AppData\Roaming\Thycotic\log`" on the machine you are launching the Protocol Handler from.
7. Copy and save the `SS-RDPWin.log` log file.
8. Return `log4net-rdp.config` to its original state by removing the comment out of `<!--<level value="INFO" />` and commenting out `<level value="DEBUG" />`

Distributed Engine Log

SSDE log

The **Secret Server distributed engine log** is responsible for recording distributed engine activity.

Location: `C:\Program Files\Thycotic Software Ltd\Distributed Engine\log`.

Customers using Secret Server 11.1.6 and newer can adjust audit logging directly from the UI. Please refer to "Setting the Logging Levels" on page 378 for more information.

Customers using a Secret Server version prior to 11.1.6 need to use a manual method of adjusting the audit logs:

1. Log in as an administrator on the distributed engine server.
2. Locate the `Thycotic.DistributedEngine.Service.exe.config` file in the `C:\Program Files\Thycotic Software Ltd\Distributed Engine` directory.
3. Open the file in a text editor.

Enabling Debug Mode in Distributed Engine Log Files

Please see "Enabling Debug Mode in DE Log Files" on page 380 for more details on how to proceed.

Enabling Discovery Logging

To enable discovery logging, change the `Thycotic.DistributedEngine.Service.exe.config` configuration file as follows:

```
<logger name="Thycotic.Discovery">
  <level value="VERBOSE" />
</logger>
```

RabbitMQ Log

Access the RabbitMQ logs at `C:\RabbitMq\log`.

Syslog Event List

Table: Syslog Event List

Type	Type ID	Action	Action ID	Description
AUTOEXPORT		DOWNLOADEXPORT	10164	Autoexported has been downloaded
AUTOEXPORT		EDIT	10161	Autoexport was edit
AUTOEXPORT		EXPORT	10162	Autoexport was exported
AUTOEXPORT		RUNEXPORT	10163	Run Autoexport
BACKUPCONFIGURATION	10023	EDIT	10143	The back configuration was edited
CHARACTERSET	10020	CREATE	10132	A character set was created

Type	Type ID	Action	Action ID	Description
CHARACTERSET	10020	DISABLE	10135	A character set was disabled
CHARACTERSET	10020	EDIT	10133	A character set was edited
CHARACTERSET	10020	ENABLE	10134	A character set was enabled
CONFIGURATION	5	BACKUP	10144	Database was backed up
CONFIGURATION	5	DATABASE EDIT	10122	Database configuration was edited
CONFIGURATION	5	EDIT	15	Configuration was edited
CONFIGURATION	5	UPGRADE	10121	An updated was started or completed
DISASTERRECOVERY		DATAREPLICATION	10191	Data replication started for disaster recovery
DISASTERRECOVERY		EDIT	10190	Disaster recovery was edited

Type	Type ID	Action	Action ID	Description
DISASTERRECOVERY		RUNDATAREPLICATION	10192	Run Data replication for disaster recovery
DOMAIN	10022	CREATE	10138	A domain was created
DOMAIN	10022	EDIT	10139	A domain was edited
DOMAIN	10022	SYNCHRONIZE	10145	A domain's users were synchronized
DUALCONTROL	10017	CREATE	10104	A dual control was created
DUALCONTROL	10017	DELETE	10106	A dual control was deleted
DUALCONTROL	10017	UPDATE	10105	A dual control was updated
ENCRYPTION	10012	HSM DISABLE	10067	HSM was disabled
ENCRYPTION	10012	HSM ENABLE	10066	HSM was enabled
ENCRYPTION		HSM ROTATE	10157	
ENCRYPTION	10012	KEY MANAGEMENT DISABLE	10148	Key management was disabled

Type	Type ID	Action	Action ID	Description
ENCRYPTION	10012	KEY MANAGEMENT EDIT	10147	Key management was edited
ENCRYPTION	10012	KEY MANAGEMENT ENABLE	10146	Key management was enabled
ENCRYPTION	10012	ROTATE MASTER ENCRYPTION KEY	10167	
ENCRYPTION	10012	ROTATE MASTER ENCRYPTION KEY FAILURE	10170	
ENCRYPTION	10012	ROTATE MASTER ENCRYPTION KEY RETRY	10168	
ENCRYPTION	10012	ROTATE MASTER ENCRYPTION KEY SUCCESS	10169	
ENCRYPTION	10012	ROTATE SECRET KEYS	10069	Rotate secret keys was requested
ENCRYPTION	10012	ROTATE SECRET KEYS CANCEL REQUESTED	10070	Rotate secret keys was canceled
ENCRYPTION	10012	ROTATE SECRET KEYS FAILURE	10072	Rotate secret keys failed
ENCRYPTION	10012	ROTATE SECRET KEYS SUCCESS	10071	Rotate secret keys succeeded
ENGINE	10014	ACTIVATE	10083	An engine was activated

Type	Type ID	Action	Action ID	Description
ENGINE	10014	CREATE	10082	An engine was created
ENGINE	10014	DEACTIVATE	10084	An engine was deactivated
ENGINE	10014	DELETE	10085	An engine was deleted
ENGINE	10014	OFFLINE	10151	An engine was taken offline
ENGINE	10014	ONLINE	10152	An engine was brought online
ENGINE		PROXYENDPOINTCHANGED	10180	Proxy endpoint was changed for node
EXPORTSECRETS	10003	EXPORTED	10016	Secrets have been exported from Secret Server
FOLDER	2	CREATE	7	A folder has been created
FOLDER	2	DELETE	8	A folder has been deleted

Type	Type ID	Action	Action ID	Description
FOLDER	2	EDITPERMISSIONS	14	A folder's permissions have been changed
FOLDER	2	SECRETPOLICYCHANGE	10053	A folder's secret policy was changed
GROUP	6	CREATE	10140	A group was created
GROUP	6	EDIT	10141	A group was edited
IMPORTSECRETS	10004	IMPORTED	10017	Secrets have been imported to Secret Server
IPADDRESSRANGE	7	CREATE	10109	An IP range was created
IPADDRESSRANGE	7	DELETE	10111	An IP range was deleted
IPADDRESSRANGE	7	GROUP ASSIGN	10114	A group was assigned to an IP range
IPADDRESSRANGE	7	GROUP UNASSIGN	10115	A group was removed from an IP range

Type	Type ID	Action	Action ID	Description
IPADDRESSRANGE	7	UPDATE	10110	An IP range was updated
IPADDRESSRANGE	7	USER ASSIGN	10112	A user was assigned to an IP range
IPADDRESSRANGE	7	USER UNASSIGN	10113	A user was removed from an IP range
LICENSES	10007	ADD	10119	A license was added
LICENSES	10007	DELETE	10120	A license was removed
NODE		PROXYENDPOINTCHANGED	10181	Proxy endpoint was changed for a Secret
PASSWORDCHANGER	10019	AUTHEDIT	10130	The password changer's authentication method was edited
PASSWORDCHANGER	10019	COMMANDCREATE	10128	A command in a password changer was created

Type	Type ID	Action	Action ID	Description
PASSWORDCHANGER	10019	COMMANDDELETE	10129	A command in a password changer was deleted
PASSWORDCHANGER	10019	COMMANDEDIT	10127	A command in a password changer was edited
PASSWORDCHANGER	10019	CREATE	10123	A password changer was created
PASSWORDCHANGER	10019	DISABLE	10126	A password changer was disabled
PASSWORDCHANGER	10019	EDIT	10124	A password changer was edited
PASSWORDCHANGER	10019	ENABLE	10125	A password changer was enabled
PASSWORDCHANGER	10019	SCANFIELDEDIT	10131	The fields the password changer uses from the secret was edited

Type	Type ID	Action	Action ID	Description
PASSWORDREQUIREMENT	10021	CREATE	10136	A password requirement was created
PASSWORDREQUIREMENT	10021	EDIT	10137	A password requirement was edited
RDPPROXY		DISABLED	10174	RDP Proxy was disabled
RDPPROXY		ENABLED	10173	RDP Proxy was enabled
ROLE	3	ASSIGNUSERORGROUP	10	A user or group was assigned to a role
ROLE	3	CREATE	9	A role was created
ROLE	3	DISABLEROLE	21	A role was disabled
ROLE	3	EDIT	10142	A role was edited
ROLE	3	ENABLEROLE	20	A role was enabled
ROLE	3	UNASSIGNUSERORGROUP	11	A user or group was unassigned from a role
ROLEPERMISSION	4	ADDEDTOROLE	12	A permission was added to a role

Type	Type ID	Action	Action ID	Description
ROLEPERMISSION	4	REMOVEDFROMROLE	13	A permission was removed from a role
SCRIPTPOWERSHELL	10008	CREATE	10027	A PowerShell script was created
SCRIPTPOWERSHELL	10008	DEACTIVATE	10028	A PowerShell script was deactivated
SCRIPTPOWERSHELL	10008	EDIT	10029	A PowerShell script was edited
SCRIPTPOWERSHELL	10008	REACTIVATE	10030	A PowerShell script was reactivated
SCRIPTPOWERSHELL	10008	VIEW	10031	A PowerShell script was viewed
SCRIPTSQL	10011	CREATE	10061	A SQL script was created
SCRIPTSQL	10011	DEACTIVATE	10062	A SQL script was deactivated
SCRIPTSQL	10011	EDIT	10063	A SQL script was edited

Type	Type ID	Action	Action ID	Description
SCRIPTSQL	10011	REACTIVATE	10064	A SQL script was reactivated
SCRIPTSQL	10011	VIEW	10065	A SQL script was viewed
SCRIPTSSH	10010	CREATE	10056	An ssh script was created
SCRIPTSSH	10010	DEACTIVATE	10057	An ssh script was deactivated
SCRIPTSSH	10010	EDIT	10058	An ssh script was edited
SCRIPTSSH	10010	REACTIVATE	10059	An ssh script was reactivated
SCRIPTSSH	10010	VIEW	10060	An ssh script was viewed
SECRET	10001	ACCESS_APPROVED	10044	A secret's access request was approved
SECRET	10001	ACCESS_DENIED	10045	A secret's access request was denied

Type	Type ID	Action	Action ID	Description
SECRET	10001	CACHEVIEW	10103	A secret has been viewed, but cached data was presented
SECRET	10001	CHECKIN	10025	A secret was checked in
SECRET	10001	CHECKOUT	10026	A secret was checked out
SECRET	10001	COPY	10020	A secret was copied
SECRET	10001	CREATE	10001	A secret has been created
SECRET	10001	CUSTOM_AUDIT	10038	A custom audit has been created
SECRET	10001	CUSTOM_PASSWORD_REQUIREMENT_ADDED	10046	A custom password requirement was added to a secret
SECRET	10001	CUSTOM_PASSWORD_REQUIREMENT_REMOVED	10047	A custom password requirement was removed from a secret

Type	Type ID	Action	Action ID	Description
SECRET	10001	DEACTIVATE	10002	A secret has been deactivated
SECRET	10001	DEPENDENCY_ADDED	10049	A dependency was added to a secret
SECRET	10001	DEPENDENCY_DELETED	10048	A dependency was removed from a secret
SECRET	10001	DEPENDENCYFAILURE	10008	A secret's dependency failed
SECRET	10001	EDIT	10005	A secret was edited
SECRET	10001	ERASE_COMPLETED	10166	A secret erased completed
SECRET	10001	ERASE_REQUESTED	10165	A secret erased request has been made
SECRET	10001	EXPIREDTODAY	10009	A secret is expiring today
SECRET	10001	EXPIRES01DAY	10010	A secret is expiring in one day

Type	Type ID	Action	Action ID	Description
SECRET	10001	EXPIRES03DAYS	10013	A secret is expiring in three days
SECRET	10001	EXPIRES07DAYS	10011	A secret is expiring in seven days
SECRET	10001	EXPIRES15DAYS	10012	A secret is expiring in fifteen days
SECRET	10001	EXPIRES30DAYS	10094	A secret is expiring in thirty days
SECRET	10001	EXPIRES45DAYS	10095	A secret is expiring in forty-five days
SECRET	10001	EXPIRES60DAYS	10096	A secret is expiring in sixty days
SECRET	10001	EXPIRES90DAYS	10182	Secret is expiring in 90 Days
SECRET	10001	EXPORTSECRET	10093	A secret was exported
SECRET	10001	FILESAVE	10102	A file was saved to a secret
SECRET	10001	HEARTBEATFAILURE	10007	A secret's heartbeat failed

Type	Type ID	Action	Action ID	Description
SECRET	10001	HEARTBEATSUCCESS	10032	A secret's heartbeat succeeded
SECRET	10001	HOOKCREATE	10035	A hook has been created
SECRET	10001	HOOKDELETE	10037	A hook has been deleted
SECRET	10001	HOOKEDIT	10036	A hook has been edited
SECRET	10001	HOOKFAILURE	10033	A hook has failed to initialize a PowerShell script
SECRET	10001	HOOKSUCCESS	10034	A hook has successfully initialized a PowerShell script
SECRET	10001	LAUNCH	10006	A secret was launched
SECRET	10001	PASSWORD CHANGE MAX ATTEMPTS REACHED	10068	A secret has reached the max amount of attempts to change its password

Type	Type ID	Action	Action ID	Description
SECRET	10001	PASSWORD_COPIED_TO_CLIPBOARD	10040	A secret's password was copied to the clipboard
SECRET	10001	PASSWORD_DISPLAYED	10039	A secret's password was displayed
SECRET	10001	PRECHECKIN	10158	A secret was pre-checked in
SECRET	10001	PRECHECKOUT	10154	A secret was pre-checked out
SECRET	10001	SECRETPASSWORDCHANGE	10055	A secret's password was changed
SECRET	10001	SECRETPASSWORDCHANGEFAILURE	10155	A secret password change failed
SECRET	10001	SECRETPOLICYCHANGE	10054	A secret's policy was changed
SECRET	10001	SESSION RECORDING VIEW	10019	A session recording of a launched secret was viewed

Type	Type ID	Action	Action ID	Description
SECRET	10001	UNDELETE	10003	A secret has been undeleted
SECRET	10001	VIEW	10004	A secret has been viewed
SECRET	10001	VIEWED_EDIT	10041	The secret's view option was edited
SECRET	10001	WEBPASSWORDFILL	10099	A secret was used to fill a web form
SECRETPOLICY	10009	CREATE	10051	A secret policy was created
SECRETPOLICY	10009	EDIT	10052	A secret policy was edited
SECRETSERVERSETTING		EXPORTED	10159	Secret Server setting was exported
SECRETSERVERSETTING		IMPORTED	10160	Secret Server setting was imported
SECRETTEMPLATE	10006	CREATE	10021	A secret template was created

Type	Type ID	Action	Action ID	Description
SECRETTEMPLATE	10006	CREATE SECRET ACCESS CHANGED	10108	Users that have access to create a secret of a template were changed
SECRETTEMPLATE	10006	EDIT	10022	A secret template was edited
SECRETTEMPLATE	10006	FIELD ENCRYPTED	10042	A field in a secret template was marked to be encrypted
SECRETTEMPLATE	10006	FIELD EXPOSED	10043	A field in a secret template was marked to be exposed
SECRETTEMPLATE	10006	OWNERS_MODIFIED	10107	The owners of a secret template were changed
SECRETTEMPLATE	10006	TEMPLATE COPIED FROM	10023	A secret template was copied

Type	Type ID	Action	Action ID	Description
SECURITYANALYTICSCONFIGURATION	10016	EDIT	10098	Privileged Behavior Analytic settings were edited
SITE	10013	ADDENGINE	10077	An engine was added to a site
SITE	10013	ASSIGNEDOMAIN	10091	A site was assigned to a domain
SITE	10013	CREATE	10073	A site was created
SITE	10013	DISABLE	10076	A site was disabled
SITE	10013	DISABLERDPPROXY	10178	RDP Proxy was disabled for Site
SITE	10013	DISABLESSHPROXY	10176	SSH Proxy was disabled for Site
SITE	10013	EDIT	10074	A site was edited
SITE	10013	ENABLE	10075	A site was enabled
SITE	10013	ENABLERDPPROXY	10177	RDP Proxy was enabled for Site

Type	Type ID	Action	Action ID	Description
SITE	10013	ENABLESSHPROXY	10175	SSH Proxy was enabled for Site
SITE	10013	ENGINEDOWNLOAD	10081	An engine was downloaded for a site
SITE	10013	ENGINEOFFLINE	10080	An engine was taken offline
SITE	10013	ENGINEONLINE	10079	An engine was brought online
SITE	10013	PROXYENDPOINTCHANGED	10179	Proxy endpoint was changed for a Site
SITE	10013	REMOVEDOMAIN	10092	A site was removed from a domain
SITE	10013	REMOVEENGINE	10078	An engine was removed from a site
SITECONNECTOR	10015	CREATE	10086	A site connector was created

Type	Type ID	Action	Action ID	Description
SITECONNECTOR	10015	CREDENTIALVIEW	10090	A site connector's credentials were viewed
SITECONNECTOR	10015	DISABLE	10089	A site connector was disabled
SITECONNECTOR	10015	EDIT	10087	A site connector was edited
SITECONNECTOR	10015	ENABLE	10088	A site connector was enabled
SSHPROXY		DISABLED	10172	SSH Proxy was disabled
SSHPROXY		ENABLED	10171	SSH Proxy was enabled
System Log	500	System Log	500	
TLS	10018	FAIL	10118	A TLS error occurred while connecting to a remote server
UNLIMITEDADMIN	10002	DISABLE	10015	Unlimited admin mode was disabled

Type	Type ID	Action	Action ID	Description
UNLIMITEDADMIN	10002	ENABLE	10014	Unlimited admin mode was enabled
USER	1	ADDEDTOGROUP	5	A user was added to a group
USER	1	CHALLENGE_APPLIED	10116	A PBA challenge was applied to a user
USER	1	CHALLENGE_CLEARED	10117	A PBA challenge was cleared from a user
USER	1	CREATE	1	A user was created
USER	1	DISABLE	2	A user was disabled
USER	1	EDIT	10100	A user was edited
USER	1	ENABLE	3	A user was enabled
USER	1	LOCKOUT	4	A user was locked out
USER	1	LOGIN	16	A user logged in
USER	1	LOGINFAILURE	18	A user failed to log in

Type	Type ID	Action	Action ID	Description
USER	1	LOGOUT	17	A user logged out
USER	1	OWNERS_MODIFIED	10097	A user's owner was modified
USER	1	PASSWORDCHANGE	19	A user's password was changed
USER	1	REMOVEDFROMGROUP	6	A user was removed from a group
USER	1	REMOVEPERSONALLYIDENTIFIABLEINFORMATION	10153	Users personal identifiable information was removed
USER	1	TWO FACTOR UPDATED	10101	A user's two factor settings were updated
USER	1	TWOFACTORRESET	10149	A user's two factor was reset
USER	1	TWOFACTORRESETFAILED	10150	A user's two factor authentication failed
USERAUDIT	10005	EXPIRENOW	10018	Secrets have been manually expired

System Log

The System Log is used to communicate the different events that are occurring while Secret Server is executing. It can be helpful in troubleshooting unexpected behavior. The system log can be enabled by clicking **Edit** and checking the **Enable System Log** check box on the **Administration > System Log** page.

See the "Syslog Event List" on page 347 for a list of system events.

System log parameters include:

- **Notify Administrators when System Log is Shrunk:** This setting is used to send an email to all system log administrators when the system log has been truncated. A system log administrator is any user in a role with the Administer System Log permission included.
- **Maximum Log Length:** This is the maximum number of rows to keep in the system log table in the SQL database. When it reaches that amount, it is reduced by 50%.
- **Allow viewing log files online:** If enabled, the server may parse its internal log file on demand to view online. Parsing the log file takes a very short time, typically less than 150 ms, but the log file cannot be rolled while the file is being read. (Adding new entries to the log file without rolling is unaffected.) If a log file roll is attempted during that time, then log entries may be dropped until the roll is successful.

To clear the system log of all its records, click **Clear log** under the Logs tab.

Secure Syslog and CEF Logging

Overview

Syslog and CEF are two widely used standards for logging and event management in the field of IT and network security.

Syslog is a standard protocol used for sending and receiving log and event messages between network devices, servers, and applications. It provides a structured and standardized way of capturing and forwarding system messages, allowing IT administrators to monitor and troubleshoot network and system issues more effectively. Syslog messages can contain various types of information, including system events, errors, and warnings, as well as user activity and other operational data.

CEF (Common Event Format) is a standardized log format for capturing and transmitting security-related events across various systems and devices. CEF provides a common language and format for security information and event management (SIEM) systems, allowing for more efficient and effective analysis of security events and threats.

See the "Syslog Event List" on page 347 for a complete list of events.

Secret Server and Syslog or CEF Logging

Secret Server can send a copy of important log messages to an external syslog server for added security using the following protocols:

Common Event Format (CEF) is an industry-standard format on top of syslog messages that ensures event interoperability between different platforms.

Table: Syslog Transportation Protocols

Protocol	Encrypted	Notes
UDP	No	Least reliable. User Datagram Protocol (UDP) traffic is fire-and-forget with no assurance messages are delivered and no error checking.
TCP	No	More reliable. Transmission Control Protocol (TCP) ensures messages arrive in order, missing messages are resent, and has built in error checking.
Secure TCP	Yes	Establishes a secure connection – Transport Layer Security (TLS) 1.1 or 1.2 only. Syslog Server's certificate is validated by Windows to ensure it is trusted and not revoked. Can be used with or without client certificates (configured in **Configuration > Security tab > TLS Auditing > Advanced**).

Due to the sensitive nature of Secret Server logs, we strongly recommend using Secure TCP.

Configuring a Secure TCP Syslog or CEF External Audit Server in Secret Server

Compatible Audit Servers

- syslog-ng
- Any Audit server that accepts TLS encrypted messages using the BSD syslog protocol

Special Characters in Syslogs

Secret Server uses UTF-8 encoding for syslog messages, which supports non-ASCII characters. Some syslog servers may only support ASCII. If you cannot change your server to UTF-8, your syslog messages may include error characters (often question marks) where non-ASCII characters appear.

Configuring an External Audit Server

1. Navigate to **Admin** > **Application**. The Application page appears.
2. Click the **Edit** button at the top right of the page.
3. Click to select the **Enable Syslog/CEF Log Output** check box. A syslog/CEF section expands below.

   Note: syslog/CEF may require an additional license key. To install licenses, navigate to **Admin** > **Licenses** > **Install New License**. Once installed, the license requires activation. Contact your Delinea Sales Representative with any questions.
4. Type IP address or name for the IIS server hosting the syslog/CEF server in the **Syslog/CEF Server** text box.

   You can add multiple entries, separating each with a semicolon.

5. Type the port number where the logging information will be passed (6514 is the default port for secure TCP syslog) in the **Syslog/CEF Port** text box.

   Note: Secret Server requires outbound access to this server and port so communication can pass freely.
6. Click the **Syslog/CEF Protocol** dropdown list and select **Secure TCP**. Secure TCP means either TLS v1.2 or v1.1 because other versions of SSL, such as SSL v3 and TLS v1.0, have known weaknesses.
7. Click to select **Syslog/CEF Time Zone** list box to **UTC Time** or **Server Time**, depending on your preference.
8. Click the **Syslog/CEF DateTime Format** dropdown to format Syslog timestamps. The standard for Syslog is ISO timestamps; however, some still use the legacy format. Syslog is the default for upgrades to allow current configurations to retain their behavior, and ISO format is the default in new instances. Syslog format: Jun 23 2022 11:22:33. ISO 8601 format: 2022-06-23T11:22:33.000. You must enable the configuration preview to modify this setting.
9. Click the **Syslog/CEF Site** dropdown to select the related Site that Syslog/CEF will run on.
10. Click to select the **Write Syslogs As Windows Events** check box to write audits and event subscriptions to the Windows Event Log of the server.
11. Click the **Save** button.

Caching Syslog Audits and the Syslog Circuit Breaker

Enabling Secure Syslog Logging

Once secure syslog logging is enabled in Secret Server, ensure that the system is set to cache syslog failure notification messages in the Secret Server database if the connection to the external syslog server breaks.

Configuring Syslog Server for Non-Local Sites

If the syslog server uses a non-local site, implement a circuit breaker system to avoid overwhelming the distributed engine and message queues during likely failure scenarios.

Understanding Circuit Breaker States

- The circuit breaker enters the "open" state after ten failures within five minutes.
- In the "open" state, the system will not attempt to send syslog messages.
- Every five minutes, the circuit breaker shifts to the "half-open" state to attempt recovery by sending a limited number of syslog messages.
- Successful delivery of three syslog messages transitions the circuit breaker to the "closed" state, resuming normal syslog message processing.

Monitoring System Alerts

When the circuit breaker is in "open" or "half-open" mode, administrators will receive a notification banner. This banner provides a link to the system log for accessing diagnostic messages, including details on which engines had issues sending messages to the syslog server and other diagnostic information.

Configure Auditing for TLS Connections

To track problems with TLS connections (including whenever the connection fails), enable the TLS certificate chain policy and error auditing in Secret Server:

1. Navigate to **Admin** > **Configuration**.
2. Click the **Security** tab.
3. Click the **Edit** button at the bottom of the page.
4. Scroll to the **TLS Auditing** section.
5. Ensure the **Apply TLS Certificate Chain Policy and Error Auditing** check box is enabled. If not, you cannot use client certificates.

> If secure TCP is used for the syslog/CEF protocol and there are one or more client certificate thumbprints entered, Secret Server checks the local computer's Web hosting and personal certificate store and uses the first one it finds.

Adding Client Certificate Thumbprints

1. Navigate to **Admin** > **Configuration**.
2. Click the **Security** tab.
3. Click the **Edit** button at the bottom of the page.
4. Scroll to the **TLS Auditing** section.
5. Click the **Advances (not required)** link. A client certificate thumbprint section appears.
6. Copy and paste a list of SHA1 SSL certificate thumbprints into the **Client Certificate Thumbprints(s)** text box. Separate each thumbprint (40 characters each) with a semicolon. Up to ten are allowed.

> Secret Server's IIS application pool must be granted permission to use the client certificates, using the Windows HTTP Services Certificate Configuration Tool (WinHttpCertCfg.exe). See Compatibility Notes for Client Certificates.

Determining the Status of a Remote Audit Server

To view the logs for any TLS-Connection related errors, perform the following:

1. Open the **Microsoft SQL Server Management Studio**.
2. Navigate to your SecretServer database at **<DB Machine Name>** > **Databases** > **SecretServer**).
3. Set up a new query.
4. Type and enter `select from tbSecurityAuditLog` to view the events from the TLS audit.

> For more detailed troubleshooting reporting, reference the logs on the Secret Server Web server at `C:\inetpub\wwwroot\SecretServer\log`). View the `SS.log`, `SS-BSSR.log` (background scheduler), and `SS-BSWR.log` (background worker) for any errors.

Compatibility Notes for Client Certificates

IIS Application Pool Certificate Permissions

> Applicable only to Secret Server Cloud

Secret Server's IIS application pool must be granted permission to use the client certificates, using the Windows HTTP Services Certificate Configuration Tool (WinHttpCertCfg.exe).

For example: `winhttpcertcfg.exe -g -c LOCAL_MACHINE\MY -s "Certificate Subject" -a "HOSTNAME\IIS APPPOOL\SecretServer"`

You can download the tool at:

Windows HTTP Services Certificate Configuration Tool (WinHttpCertCfg.exe)

You can view the documentation at:

WinHttpCertCfg.exe, a Certificate Configuration Tool

Otherwise, if Secret Server is configured to use a client certificate, and IIS does not have permission, errors like this may appear in the logs:

TLS Error Detected (Authentication Error connecting to IP:PORT) - The credentials supplied to the package were not recognized.

If you are using a client certificate, and a syslog-ng logging server, the following message may occasionally appear in the main syslog-NG log file:

SSL error while reading stream; tls_error='SSL routines:ssl_get_prev_session:session id context uninitialized'

On the Secret Server side, this appears:

TLS Error Detected (Authentication Error connecting to IP:PORT) - Authentication failed because the remote party has closed the transport stream.

This is caused by Windows trying to cache secure connections when client certificates are used, but because syslog-ng has not configured the OpenSSL "session id context", OpenSSL displays this error when it tries to resume a previous session.

Secret Server automatically reconnects and resends any missed messages, so the errors should not have an impact. However, you can disable Window's secure connection caching by adding the ClientCacheTime setting set to 0 in the Registry and then rebooting. This did not cause any significant performance impact in internal testing.

> If changing back to a previous syslog IP address and port, you will receive a closed connection TLS error on the first attempted syslog connection after making the change. A subsequent call will succeed as the first failure will clear the cached connection on Windows. This is due to the issue with syslog-ng.

> If syslog-ng configures their OpenSSL session id context, this error message correction is no longer needed.

AlienVault

It is common for people to incorrectly use the client certificate thumbprints feature when setting up secure AlienVault for syslog. This can cause Secret Server to try to connect to LDAPS with the AlienVault certificate, which can break LDAPS. Users should not use the Secret Server client certificates thumbprint for specifying one certificate for syslog and another for LDAP. The certificate list is intended for each Secret Server or DE to have its own, unique certificate.

Giving Application Pools Event Log Access

> This topic only applies to **Secret Server On-Premises**.

Overview

When the database becomes inaccessible, Secret Server will try to log errors to the Windows event log. By default, network service and standard service accounts will not have permissions to the event log. Permissions must be added to specific event log registry keys.

Applying Windows Event Log Permissions

1. Determine the account that is running Secret Server:
   a. Log on Secret Server.
   b. Go to **Admin > Diagnostics**.
   c. Look for any of the **Thread Identity** labels. These contain the identity of Secret Server (often `NT AUTHORITY\NETWORK SERVICE` or `IIS APPPOOL\SecretServer` or the service account set up for IWA. See "Running the IIS Application Pool As a Service Account" on page 60.

      > You can also determine the identity by logging in and navigating to http://yoursecretserverurl/Installer.aspx. The first step of this page will tells you the application pool identity.

2. Open the Windows registry editor on the machine running Secret Server (regedit at the command prompt or Window search text box).
3. On the left, navigate to **HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog**.
4. Right click the **EventLog** folder in your registry editor and select **Permissions**. A permissions dialog box appears.
5. Click the **Advanced** button.
6. On the **Permissions** tab, Click the **Add** button. A Permission Entry dialog appears.
7. Click the **Select a principal** link. The Select User, Computer… dialog box appears.
8. Find the account running Secret Server, such as Thycotic_Service (svc_thycotic@test.com).
9. Click the **OK** button. The dialog box closes.
10. In the **Basic Permissions** section of the **Permission Entry** dialog, click to select the **Read** check box.

11. Click the **Show advanced permissions** link. The pane switches.
12. Click to select the **Set Value** and **Create Subkey** check boxes in the **Advanced Permissions** section.
13. Click **OK** buttons on the remaining dialogs to apply the permissions. You are returned to the main registry editor window.
14. Navigate to **HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog > Security,** right-click and select "**Permissions...**"
15. Right click **Security** folder and select **Permissions**. A permissions dialog box appears.
16. Click the **Add** button.
17. Find the account running Secret Server.
18. Click the **OK** button.
19. Click to select the **Read** check box in the Allow column.
20. Click the **OK** button to apply the permission.
21. If you are running Windows Server 2019 or above, use the same procedure to add Read permission to **HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog > State**.

You may need to run regedit as the system user when using more recent versions.

Required Registry Permissions

After setup, the following permissions appear in the registry:

HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog

Applies to key and subkeys

- Read permissions:
  - Query Value
  - Enumerate Subkeys
  - Notify
  - Read Control
- Set Value permission
- Create Subkey permission

HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog > Security

Applies to key and subkeys

Read permissions:

- Query Value
- Enumerate Subkeys
- Notify
- Read Control

HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > EventLog > State

Applies to key and subkeys. Only applies to Windows Server 2019.

Read permissions:

- Query Value
- Enumerate Subkeys
- Notify
- Read Control

Managing Full SQL Server Transaction Logs

This topic only applies to **Secret Server On-Premises**.

SQL Server maintains a history of all operations in a transaction log. If this transaction log becomes full, you may receive one or more of the following errors:

```
System.ArgumentException: Cannot add two background tasks with the same name.
```

```
Thycotic.Data.DataAccessorException: The transaction log for database '{database}' is full. To find out why space in the log cannot be reused, see the log_reuse_wait_desc column in sys.databases
```

By default, a transaction log can grow to an unrestricted size, but some may become full in the following circumstances:

- The drive where the transaction log file is kept is out of disk space.
- The transaction log file hits its growth limit.

Potential Solutions

- Back up the log.
- Free up disk space so that the log can grow automatically.
- Move the log file to a disk drive with sufficient space.
- Increase the size of the log file.
- Add a log file on a different disk.
- Complete or kill a long-running transaction.
- Switch to simple recovery mode and truncate the log.

For more detailed information on transaction logs in SQL, see Manage the size of the transaction log file.

Setting the Logging Levels

Overview

Secret Server Web nodes and distributed engines log levels are remotely configurable and collectable. This feature is especially useful for large systems with many nodes and engines.

Configuration for the web nodes is found on the Server Nodes configuration page, alongside role settings. Configuration for distributed engines is found in the Distributed Engine configuration page. Log levels include: All, Debug, Info, Warn, Error, Off, and Not Set (the default). Previously, manual configuration file changes were required. "Not Set" relies on the configuration files for the logging level, which was the previous default behavior.

The diagnostic feature for collecting logs is improved and now gathers logs from all nodes and engines.

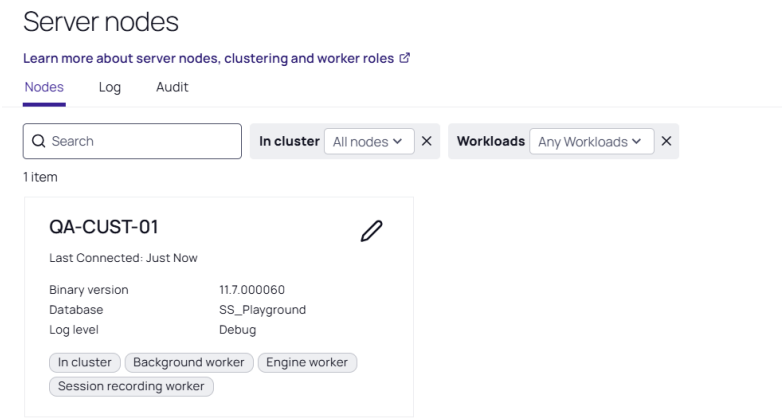
This feature is at **Admin > Server Nodes and Admin >**

Setting the Logging Level

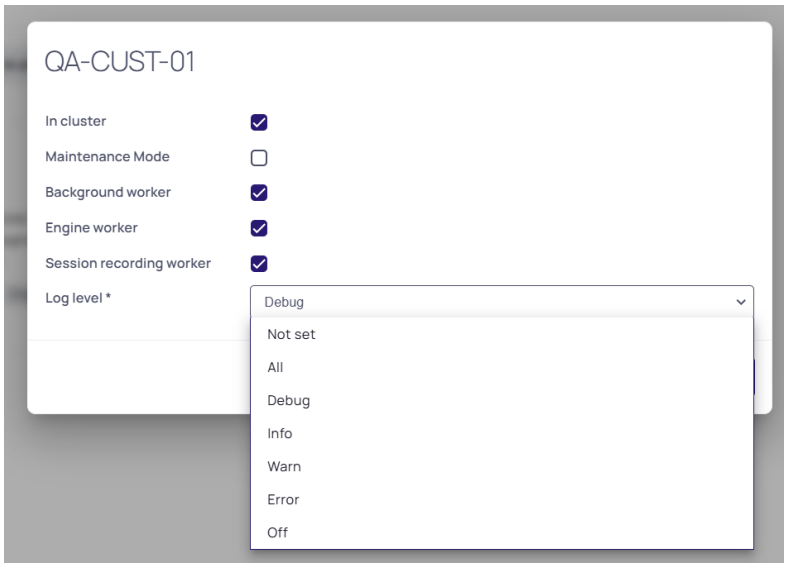
Secret Server Web Servers

For Web servers, you can set the log Level for web server nodes from the Server Nodes page:

1. Go to **Admin > Server Nodes**. The Server Nodes page appears:



2. Click the edit icon next to the server you want to change. The settings become editable.
3. Click the **Log Level** dropdown list to select the desired level.

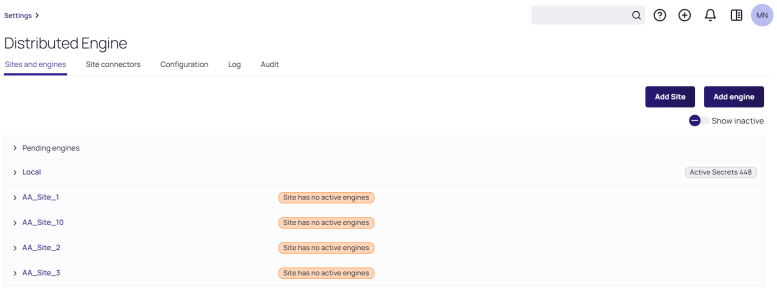


4. Click the save icon on the far right to save the change.

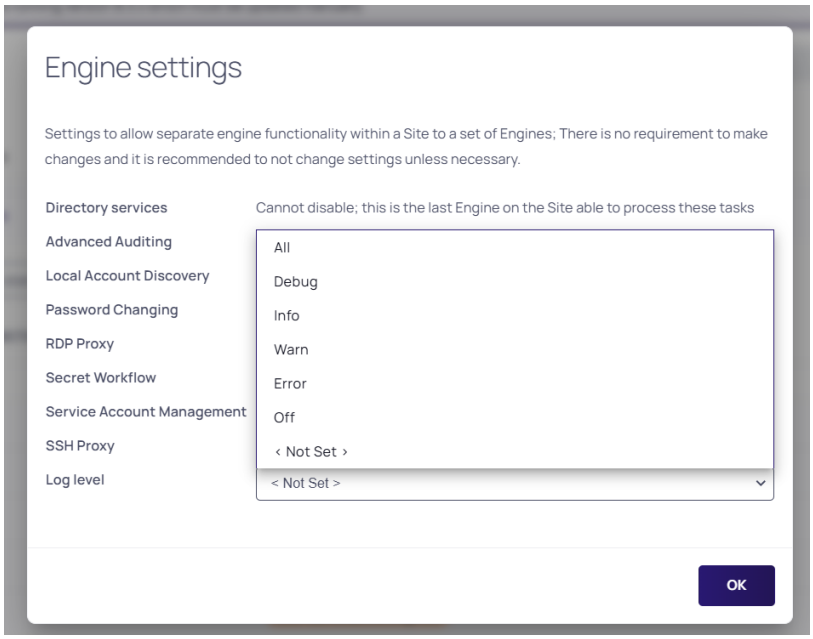
Distributed Engines

Similarly, for distribution engines:

1. Go to **Admin > Distributed Engine**. The Distributed Engine page appears:



2. Hover the mouse pointer over the desired engine. Three dots appear on the far right.
3. Click the dots icon and select **Show Settings**. The Engine Settings popup page appears:
4. Click the **Log Level** dropdown list to select the desired level.
5. Click the **OK** button to save the change.



Setting the Logging Level when a Node is Down (No Access to UI)

To manually change the logging level when a web node is down and there is no access to the UI, you can follow these steps:

1. Locate the Configuration File:
   - Access the server where Secret Server is installed.
   - Navigate to the web application's root directory, typically found at *C:\inetpub\wwwroot\SecretServer*.
2. Edit the *web-log4net.config* file:
   - Open the *web-log4net.config* file in a text editor.
   - Use the find function (Ctrl + F) to locate the `<log4net>` section.
   - Modify the Logging Level:
     - Find the line with `<level value="INFO" />` or the current logging level.
     - Change the value to the desired logging level, such as `<level value="DEBUG" />`.
     - Ensure that the line is uncommented by removing any surrounding `<!-- and -->`.
3. After making the changes, restart the IIS service to apply the new logging configuration. This can be done by running `iisreset` in the command prompt.
4. Verify the Changes:
   - Check the logs to ensure that the new logging level is capturing the desired information.

   See Enabling Debug Mode in System Logs for more details.

Enabling Debug Mode in DE Log Files

This topic discusses enabling DEBUG mode for distributed engine logs for troubleshooting.

Overview

You can expand Secret Server logging capability to locate additional information regarding an error or to help with troubleshooting an issue.

Procedure

How to enable DEBUG logging mode:

1. Log in as an administrator on the distributed engine server.
2. Locate the `Thycotic.DistributedEngine.Service.exe.config` in the `C:\Program Files\Thycotic Software Ltd\Distributed Engine` directory.
3. Open the file in a text editor.
4. Run a find (**<Control>+ <F>**) command.
5. Type in `log4net` and press **<Enter>** to locate that section, which is usually at the top.
6. Locate the lines that contain "INFO".
7. Replace each "INFO" with "DEBUG".
8. Restart the Delinea Distributed Engine service to apply the log configuration change.
9. After DEBUG mode is enabled in the system log, you can reproduce the issue, investigate the error, or send the updated logs in with your support case.

Verbose Mode

On occasion, you may be instructed to enable the VERBOSE mode to capture details for troubleshooting. Do this by using the same procedure as above but replace "INFO" or "DEBUG" with "VERBOSE" instead.

> Enabling VERBOSE mode will create very detailed log information with large numbers of log files that can accumulate and quickly consume available resources on the machine. Therefore, you should only enable it during the troubleshooting process and immediately turned it off afterward in order to prevent performance issues.

Secret Server Authentication and Authorization

Secret Server provides integration options for Windows authentication and SAML to automatically authenticate users to the application when they browse to Secret Server on their workstations. Secret Server also allows you encrypt data at various locations.

IWA Overview

Integrated Windows Authentication (IWA) is a Microsoft protocol used for user authentication in web services and applications. It leverages the credentials of logged-in Windows users to authenticate them automatically without prompting for a username or password. This mechanism is particularly beneficial in corporate environments where users access multiple services frequently and streamlines the authentication process while enhancing security.

Key Features of IWA Webservices

- Single Sign-On (SSO): IWA allows users to log in once with their Windows credentials and gain access to multiple applications without needing to re-enter their credentials. This feature significantly improves user experience and productivity.
- Security: By using Kerberos or NTLM (Windows challenge/response) protocols, IWA provides robust security. Kerberos is preferred due to its stronger encryption and mutual authentication capabilities, but NTLM is used for compatibility with older systems.
- Seamless Integration: IWA integrates seamlessly with Active Directory (AD), enabling organizations to manage user identities and permissions centrally. This integration ensures that security policies are consistently enforced across all applications.
- Reduced Administrative Overhead: With IWA, there is less need for maintaining separate authentication systems or databases for different applications. This consolidation reduces administrative overhead and potential points of failure.
- Support for Modern Web Applications: IWA is supported by various modern web servers and browsers, including Internet Information Services (IIS), Google Chrome, and Microsoft Edge. This broad compatibility ensures that it can be used in diverse IT environments.

Typical Use Cases

- Intranet Applications: IWA is ideal for intranet applications where all users are within the same Windows domain.
- Corporate Portals: It can be used to authenticate users accessing corporate portals that aggregate multiple services.
- Web Services: Developers can leverage IWA to secure web services that need to authenticate users against Active Directory.

Implementation Considerations

- Browser Configuration: For IWA to work, browsers must be configured to allow automatic logins. This typically involves setting trusted sites or intranet zones.
- Kerberos vs. NTLM: While Kerberos is more secure, NTLM may be necessary for compatibility reasons. The choice depends on the specific environment and security requirements.
- Network Topology: IWA works best in environments where users and services are within the same network or domain. Cross-domain or internet-based access might require additional configurations or different authentication mechanisms.

For details on how IWA works and best practices for its implementation, see Windows Authentication Overview.

Integrated IWA

Secret Server also provides a Integrated Windows Authentication (IWA) webservice that uses IWA instead of a username and password. This webservice can be used in an application or script to access Secret Server and retrieve secrets with storing the login credentials in the application or configuration file.

> See "Configuring Integrated Windows Authentication" below for more advanced technical information on using this webservice.

Configuring Integrated Windows Authentication

> This article applies to Secret Server On-Premises only.

Introduction

Integrated Windows Authentication (IWA) allows users to log into Secret Server automatically if they are logged into a workstation with their Active Directory credentials.

> To ensure users can access the Health Status page see "Checking Secret Server Site Status" on page 764 without logging in, be sure to enable Form and Anonymous authentication in IIS.

> "Secure LDAP" on page 520 only works with Integrated Windows Authentication in Server 2008 R2 and later.

> **Important:** The SDK is designed to be used as it is shown below and is not designed to be run using IWA to retrieve tokens or Secret information. Given this, the SDK is not supported with IWA.

Setting Up Windows Authentication

Task 1: Configuring Secret Server

1. Log into Secret Server as a user with Active Directory administration privileges.
2. Navigate to and click **Administration > Directory Services** (In the General section). The Directory Services page appears, opened to the Domains tab.
3. Click the **Add Domain** button and select **Active Directory Domain** or **Azure Active Directory Domain**. An Active Directory popup appears.
4. If you chose Active Directory Domain:
   a. Type your FQDN in the **Fully Qualified Domain Name** text box.
   b. Type the name for people to read in the **Friendly Name** text box.
   c. Click to select the **Active** check box.
   d. If you wish to use secure LDAP, click to select the **Use LDAPS** check box.
   e. Click the **Create New Secret** link under **Synchronization Secret**. A popup appears.
   f. Click **Active Directory** Account in the Choose a Secret Template dropdown list. A Create New Secret popup appears.
   g. Fill in the popup with your desired parameters for your AD secret.
   h. Click the **Create Secret** button. The popup disappears and the secret name appears on the previous popup.

i. Click the **Site** dropdown list to select your desired site.

j. Click the **Multifactor Authentication** dropdown list to select your desired type of MFA, if any.

k. Click the **Validate & Save** button to commit your choices. The popup disappears and your directory service appears in the table on the Directory Services page.

5. If you chose Azure Active Directory Domain:

   a. Type your name for people to read in the **Domain Name** text box.

   b. Click to select the **Active** check box.

   c. Type your tenant ID in the **Tenant ID** text box.

   d. Type your client ID in the **Client ID** text box.

   e. Type your client password in the **Client Secret** text box.

   f. Click the **Multifactor Authentication** dropdown list to select your desired type of MFA, if any.

   g. Click the **Validate & Save** button to commit your choices. The popup disappears and your directory service appears in the table on the Directory Services page.

6. Click the **Configuration** tab.
7. Click the **Edit** button in the **Directory Services** section. The section becomes editable.
8. If necessary, click to select the following check boxes:
   - **Enable Directory Services**
   - **Enable Integrated Windows Authentication**.
9. Click the **Save** button.
10. Click the **Edit** button in the **User Synchronization** section. The section becomes editable.
11. Ensure the **Enable User Synchronization** check box is selected.
12. Type the in the **Days**, **Hours**, and **Minutes** text boxes to choose a synchronization interval, which is how often Secret Server pulls in users from AD.
13. Select your desired option from the **User Account Options** dropdown list.
14. Select how to handle inactive users in the **Automatic User Management** dropdown list.
15. Type your desired number of days in the **Days to Keep Operational Log** text box.
16. Click the **Save** button. The Active Directory Configuration page returns to being read only.
17. Click the **Domains** tab.
18. Click the **Synchronize Now** button. This pulls all the users of the specified groups into Secret Server.

Task 2: Configuring IIS

1. Start the Internet Information Services (IIS) Manager.
2. Navigate to and select your Secret Server website in the **Connections** tree:
3. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.

4. Enable the **Windows Authentication** parameter by right-clicking it and selecting **Enable**. For now, ignore the alert if it appears in the Alert section.

   If Windows Authentication is not visible, ensure that the Windows Authentication Role service is enabled in Windows. This is different than earlier versions.

5. Disable the **Anonymous Authentication**.
6. Disable the **Forms Authentication**. The alert in the Alert section should disappear.
7. When finished, the Authentication settings should only have Windows Authentication enabled.
8. Restart your IIS server with an `iisreset` command.
9. On the Secret Server folder, ensure users have read or higher permission, and ensure the security settings are set to be inherited by child objects. Because Secret Server impersonates those users, they require access to Secret Server files.
10. Log in to the Secret Server site from an authenticated workstation.

Task 3: Configuring Secret Server Launchers

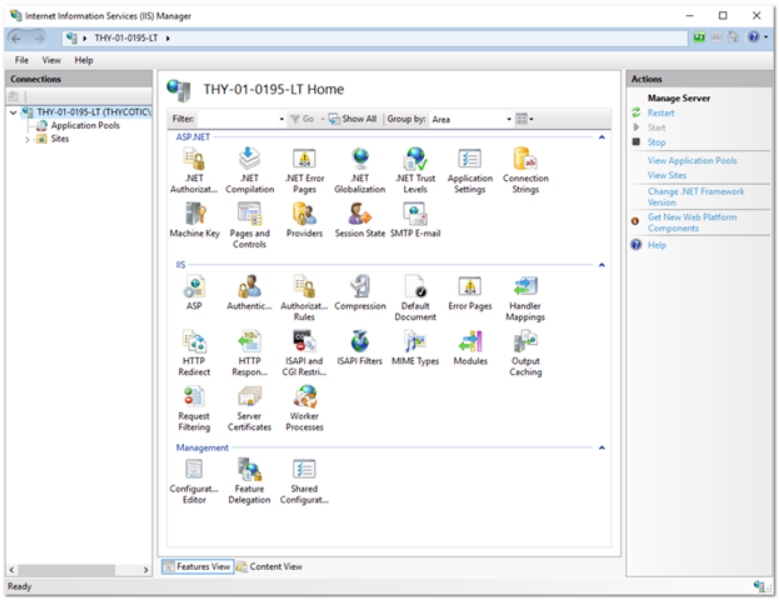
By default, a launcher will not work when using IWA, resulting in an HTTP 401: Unauthorized error. If this is an issue, ensure Secret Server is on Windows Server 2008 or later and complete the following steps:

1. Open IIS and browse to your Secret Server application.
2. Click the **>** to see the application's folders.
3. Click to select the **launchers** folder. The launchers Home panel appears.
4. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.
5. Ensure the **Anonymous Authentication** is set to **Enabled**.
6. Ensure the **Windows Authentication** is set to **Disabled**.
7. Ensure all others are disabled. When you are finished, the settings should have Anonymous Authentication enabled.
8. Click the **webservices** folder.
9. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.
10. Ensure the **Anonymous Authentication** is set to **Enabled**.
11. Ensure the **Windows Authentication** is set to **Disabled**.
12. Ensure all others are disabled. When you are finished, the settings should have Anonymous Authentication enabled
13. Click the **rdp** folder.
14. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.
15. Ensure the **Anonymous Authentication** is set to **Enabled**.
16. Ensure the **Windows Authentication** is set to **Disabled**.
17. Ensure all others are disabled.

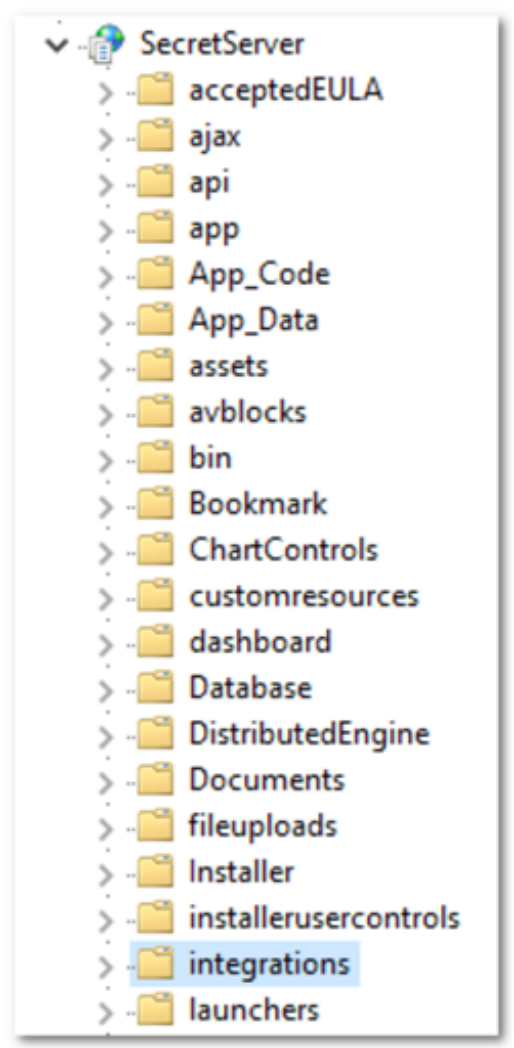
Task 4: Configuring Distributed Engines

Similarly, Secret Server with distributed engines will not work with IWA by default. If this is an issue, complete the following:

1. In Windows Explorer, navigate to the …\SecretServer\ folder. This folder is mapped to your SecretServer folder in your webserver.
2. Create a subfolder named …\SecretServer\integrations.
3. Create a subfolder called …\SecretServer\api in the same location.
4. In your …\SecretServer\api folder, create a subfolder named …\SecretServer\api\DistributedEngine.
5. Start IIS Manager:



6. Navigate the **Connections** tree back to **integrations** folder in the **SecretServer** node:

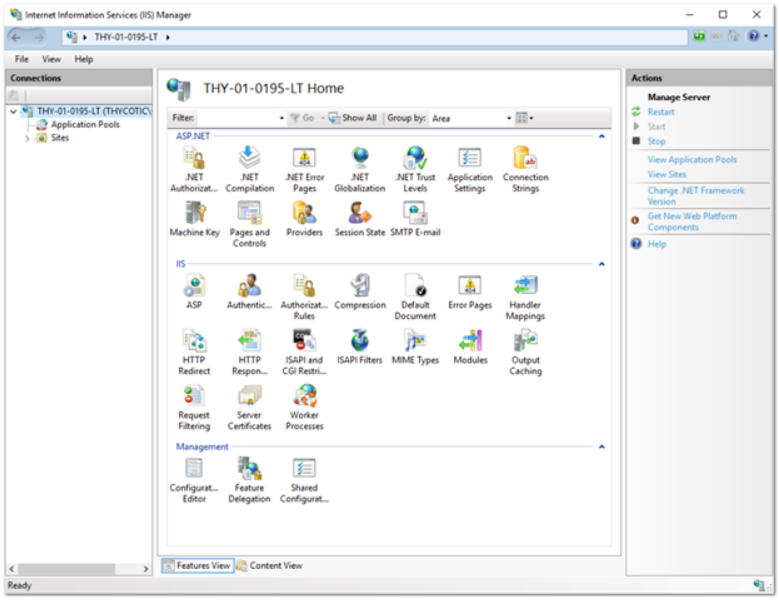


7. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.
8. Ensure the **Anonymous Authentication** is set to **Enabled**.
9. Ensure the **Windows Authentication** is set to **Enabled**.
10. Ensure all others are disabled.
11. Navigate to the …`\SecretServer\api\DistributedEngine` folder.
12. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.
13. Ensure the **Anonymous Authentication** is set to **Enabled**.
14. Ensure the **Windows Authentication** is set to **Disabled**.
15. Ensure all others are disabled.

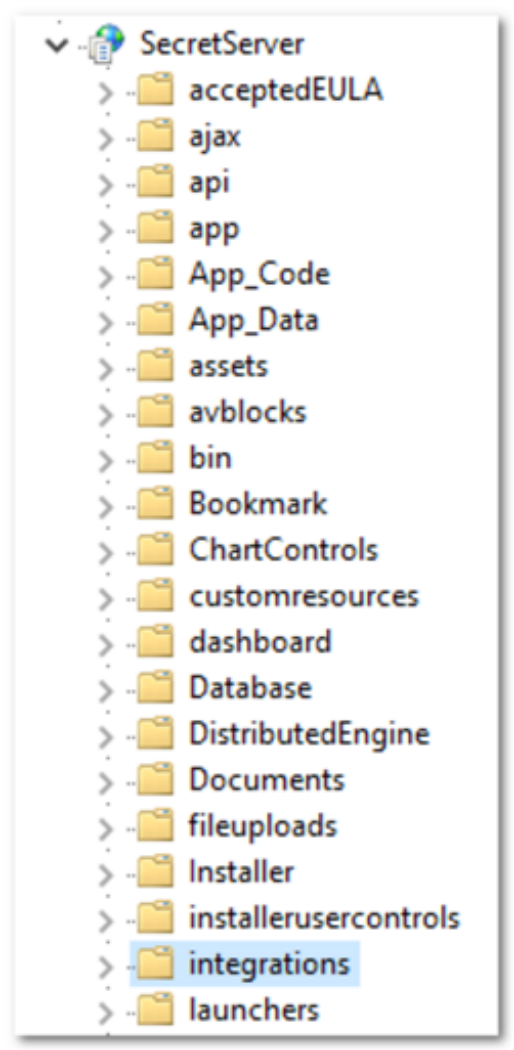
Task 5: Configuring Disaster Recovery

Similarly, Secret Server Disaster Recovery will not work with IWA by default. If this is an issue, complete the following:

1. In Windows Explorer, navigate to the …\SecretServer\ folder. This folder is mapped to your SecretServer folder in your webserver.
2. Create a subfolder named …\SecretServer\integrations.
3. Start IIS Manager:



4. Navigate the **Connections** tree back to **integrations** folder in the **SecretServer** node:



5. Double-click the **Authentication** icon in the **IIS** section to open the **Authentication** pane.
6. Ensure the **Anonymous Authentication** is set to **Enabled**.
7. Ensure all others are disabled.

Task 6: Configuring Client Certificates

If you are using client certificates, configure the following in IIS for launchers to work:

1. Click to select the **launchers** folder. The launchers Home panel appears.
2. Double-click the **SSL Settings** icon. The settings panel appears.
3. Click to set the **Client Certificates** selection button to **Accept**.
4. A dialogue box requiring a **yes** response pops up.
5. Click **Yes**.
6. Click to select the **Webservices** folder.

7. Once again, double-click the **SSL Settings** icon.
8. This time, set the **Client Certificates** selection button to **Ignore**.

If you are not automatically logged in to Secret Server after setting up IWA, IIS may not be handling the credentials correctly. To fix this, recreate the web site in IIS.

When testing IWA, keep in mind the requirements at Your Browser May Prompt You for a Password.

You may not be able to log in using IWA on the server running Secret Server for Server 2008 or later because of security settings.

Troubleshooting

AD User Prompted for Credentials Even Though IWA Is Active

A user is logged onto their machine with the same Active Directory credentials they can log into Secret Server with, but the browser still prompts them for their credentials to reach the site. Ensure your Secret Server site is included in a security zone that allows for automatic logon:

1. In your browser, go to Internet **Options > Security**.
2. Click the **Trusted Sites** security zone.
3. Click the **Custom Level** button. The Security Settings – Trusted Sites Zone dialog box appears.
4. Scroll down to **User Authentication**.
5. Click to select the **Automatic logon with current user name and password** selection button.
6. Click the **OK** button.

Logging in as a Local Account Is Not Available

In Secret Server 10.0 and later, Secret Server requires Integrated Mode in IIS. The Integrated Mode can only support either Window Authentication or Forms Authentication (used for local account authentication), not both. Because of this limitation, Forms Authentication must be disabled for the site when using Integrated Windows Authentication. Thus, logging in as Secret Server local account is not available when IWA is enabled.

Installing Windows Authentication in Windows Server 2012 Manager

1. In Server Manager, click the **Manage** menu and select **Add Roles and Features**. The Add Roles and Features wizard appears.
2. Click the **Next** button. The Select installation type window appears.
3. Select the installation type.
4. Click the **Next** button. The Server selection window appears.
5. Select the destination server.
6. Click the **Next** button. The Server roles window appears.
7. Click to expand **Web Server (IIS) > Web Server > Security**.
8. Click to select **Windows Authentication**.

9. Click the **Next** button. The Select features window appears.
10. Click the **Next** button. The Confirmation window appears.
11. Click the **Install** button. The Results window appears.
12. Click the **Close** button.

Using Webservices with IWA via Perl

Overview

You can enable webservices at **Admin > Configuration** on the **General** tab. Checking the **Enable Webservices** check box makes the ASP.NET SOAP and REST webservices built into Secret Server available for use. Additional steps are needed in IIS to ensure proper access.

Integrated Windows Authentication (IWA) does **not** work on Secret Server Cloud.

This procedure only works if Secret Server on-premises is installed on IIS 7 or greater.

Procedure

To enable IWA for webservices in IIS:

1. Open IIS Manager (`inetmgr`).
2. Expand the **Sites** node until you locate your Secret Server application or website
3. Expand the **Secret Server** node to locate the **winauthwebservices** folder.
4. Click on the **winauthwebservices** folder.
5. Click on **authentication** in the **Security** section.
6. Disable **Anonymous Authentication**.
7. Enable **Windows Authentication**.

If you are using IIS7 or greater and do not see this option, the option will need to be added through the server roles (webserver). IIS may give an alert about using both challenge and redirect-based authentication, which you can ignore.)

8. Open Windows Explorer.
9. Navigate to the **winauthwebservices** folder.
10. Give **read access** to the **winauthwebservices** folder to the domain users and groups that will be using IWA to access the webservices.

Example

Overview

The SOAP web service URL for IWA is `<Secret Server URL>/winauthwebservices/sswinauthwebservice.asmx`.

The method below uses the `SecretServerGetSecret.ps1` PowerShell script to make the SecretGet WebService call, exposing it through the `SecretServer.pm` Perl package. The `Sample.pl` file uses the `SecretServer.pm` package to retrieve specific fields from the result.

The flow is as follows:

1. Your Perl script (`sample.pl`) makes a request to the `SecretServer.pm` package.
2. The `SecretServer.pm`package passes the request on to the `SecretServerGetSecret.ps1` PowerShell script.
3. The `SecretServerGetSecret.ps1` PowerShell script calls the Secret Server web services and authenticates using the service account that `sample.pl` is running under.
4. The results are passed back to `SecretServer.pm` and then on to your Perl script (`Sample.pl`)
5. Create the following three files:

SecretServerGetSecret.ps1

```
# Sample Powershell Script
# demonstrating retrieval of a Secret from <Secret Server URL />
# via web service protected by Windows Authentication
# returned as Xml
$where = $args[0]
$secretId = $args[1]
$ws = New-WebServiceProxy -uri $where -UseDefaultCredential
$wsResult = $ws.GetSecret($secretId)
$res = convertto-xml $wsResult.Secret -As string -Depth 20
$res
```

SecretServer.pm

```
package SecretServer;
use strict;
sub usage {
    print "\nUsage: GetSecret [webservice url] [secretid]\n";
}
sub new {
    my($class, %args) = @_;
    my $self = bless({}, $class);
    return($self);
}
sub get_secret {
    my($self, $url, $secretid) = @_;
    my $result = powershell.exe .\\SecretServerGetSecret.ps1 $url $secretid;
```

```
    return($result);
}
sub get_field_from_result {
    my($self, $result, $field) = @_;
    $result =~/<Property Name="Value" Type="System.String">([^<>]+)<\/Property>
(?:\s*<Property Name="(?!FieldName)[^"]+"[^>]+>[^<]+<\/Property>\s*)*<Property
Name="FieldName"[^<>]+>$field<\/Property>/gsi;
    return("$1");
}
1;
# this is if you want to execute the Get Secret call manually from the command line
# if (@ARGV != 2)
# {
  # usage();  # Call subroutine usage()
  # exit();   # When usage() has completed execution,
            # # exit the program.
# }
# my $url = $ARGV[0];
# my $secretid = $ARGV[1];
# my $result = powershell.exe .\\SecretServerGetSecret.ps1 $url $secretid;
# print $result;
```

Sample.pl

```
use lib 'C:/<Path to the SecretServer.pm file>';
use SecretServer;
my $x = SecretServer->new();
# Change this value to match your URL
my $url = '<Secret Server URL> /winauthwebservices/sswinauthwebservice.asmx';
# Change this value to match your desired Secret Id
my $secretid = 17;
my $result = $x->get_secret($url, $secretid);
my $username = $x->get_field_from_result($result, 'UserName');
my $password = $x->get_field_from_result($result, 'Password');
print "$username : $password";
```

Using Webservices with IWA via PowerShell

Overview

You can enable webservices at **Admin > Configuration** on the **General** tab. Checking the **Enable Webservices** check box makes the ASP.NET SOAP and REST webservices built into Secret Server available for use. Additional steps are needed in IIS to ensure proper access.

> Integrated Windows Authentication (IWA) does **not** work on Secret Server Cloud.

> This procedure only works if Secret Server on-premises is installed on IIS 7 or greater.

Procedure

To enable IWA for webservices in IIS:

1. Open IIS Manager (`inetmgr`).
2. Expand the **Sites** node until you locate your Secret Server application or website
3. Expand the **Secret Server** node to locate the **winauthwebservices** folder.
4. Click on the **winauthwebservices** folder.
5. Click on **authentication** in the **Security** section.
6. Disable **Anonymous Authentication**.
7. Enable **Windows Authentication**.

   If you are using IIS7 or greater and do not see this option, the option will need to be added through the server roles (webserver). IIS may give an alert about using both challenge and redirect-based authentication, which you can ignore.)

8. Open Windows Explorer.
9. Navigate to the **winauthwebservices** folder.
10. Give **read access** to the **winauthwebservices** folder to the domain users and groups that will be using IWA to access the webservices.

Access Examples

SOAP

The SOAP web service URL for IWA is `<Secret Server URL>/winauthwebservices/sswinauthwebservice.asmx`.

Example script:

```
# Sample Powershell Script
# demonstrating retrieval of a Secret from <Secret Server URL />
# via web service protected by IWA
$where = 'http://mysecretserver/winauthwebservices/sswinauthwebservice.asmx';
$secretId = 1
$ws = New-WebServiceProxy -uri $where -UseDefaultCredential
$wsResult = $ws.GetSecret($secretId, $false, $null)
if ($wsResult.Errors.length -gt 0){
    $wsResult.Errors[0]
}
else
{
    $wsResult.Secret
}
```

REST

REST web service references the same winauthwebservices folder as SOAP when doing IWA, but in code the URL endpoint does not need to change.

Example script:

```
# Sample Powershell Script
# demonstrating authentication via web service protected by IWA
$api = 'http://mysecretserver/winauthwebservices/api/v1';
$endpoint = $api'/secrets/3844'
$secret = Invoke-RestMethod $endpoint -UseDefaultCredentials
```

OAuth

OAuth is an open-standard authorization framework that enables secure, delegated access to protected resources on the web. It allows a client application to access a user's resources on another service provider's website without sharing their login credentials. Instead, the client application requests an access token, which is granted by the service provider after the user authenticates and authorizes the request. The access token is then used to authenticate and authorize subsequent requests to the protected resources, allowing the client application to access the user's data without exposing their sensitive login information. OAuth provides a standardized, secure, and flexible way to manage access to APIs, enabling users to control how their data is shared and used by third-party applications.

OAuth in Secret Server provides a secure and efficient framework for token-based API authentication, enabling seamless integration with various applications and services. By leveraging OAuth, Secret Server allows for the delegation of access rights without sharing credentials, enhancing security and simplifying the management of permissions. This is particularly useful for enabling single sign-on (SSO) and integrating with identity providers like OpenID Connect, which is layered on top of OAuth 2.0. The implementation of OAuth in Secret Server supports multi-factor authentication (MFA) and conditional access policies, ensuring robust security measures are in place for accessing sensitive data and performing privileged operations.

Enabling Refresh Tokens for Web Services

Overview

Many modern secure applications use access tokens to ensure that users have access to the resources appropriate for them. Access tokens typically have a limited lifetime to ensure that information they contain or reference doesn't become stale, and to limit the time available for an attacker to use a stolen token.

When an access token expires or becomes invalid but the application still needs to access a protected resource, the application must use a new access token. To provide a new access token without requiring the user to grant permission a second time, OAuth 2.0 introduced an artifact called a *refresh token*.

Note the following:

- You cannot use a refresh token more than once.
- You cannot use a refresh token if your API Session Timeout is set to `unlimited`.

- In Secret Server, the refresh token "Time to live" equals the APISessionTimeout plus 15 minutes.
- Access tokens retrieved from REST can also be used for SOAP.

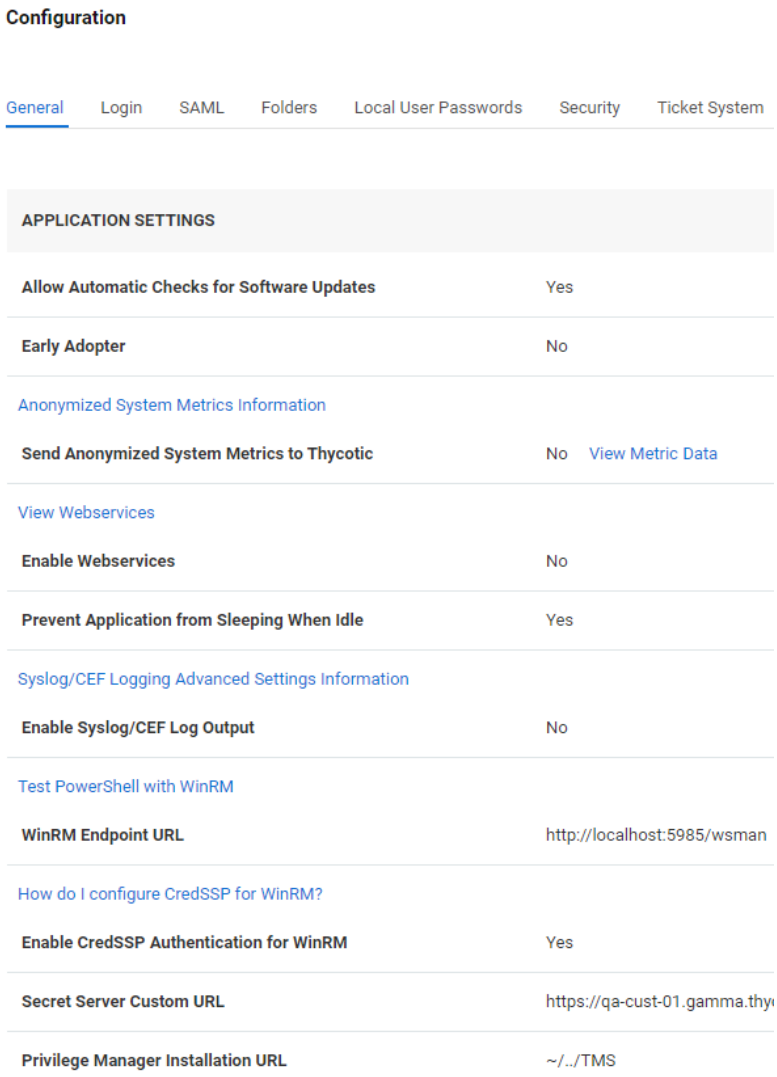
How to Enable Refresh Tokens in Secret Server

Procedure

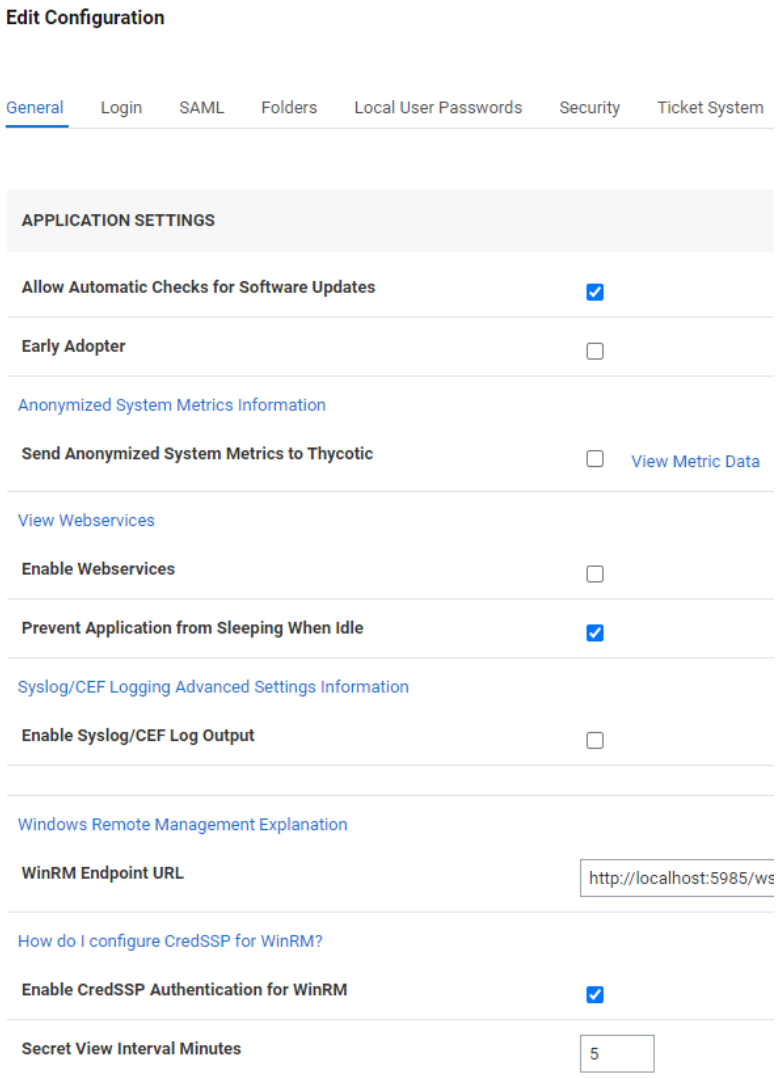
You will receive a refresh token only if the option is enabled in **Admin** > **Configuration** as described below.

1. Click **Admin** > **Configuration** > then click the **General** tab.

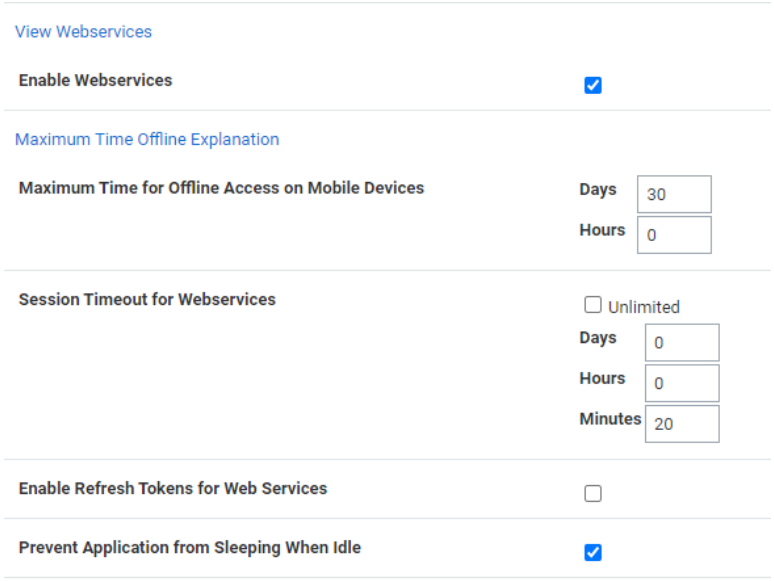
   The **Enable Web Services** field is visible but not editable.



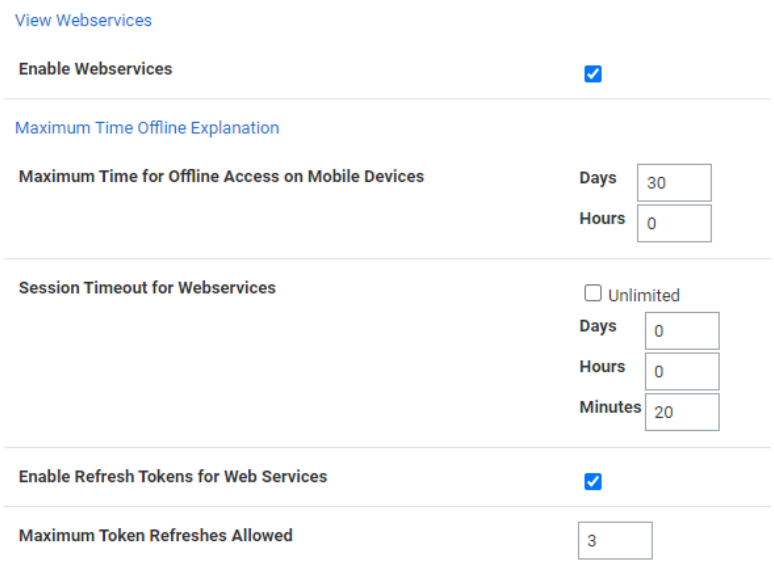
2. Scroll to the bottom of the window, click the **Edit** button, and scroll back up. The window title changes from **Configuration** to **Edit Configuration** and the **Enable Web Services** field is now editable.



3. Check the box next to **Enable Web Services**. The menu expands and the **Enable Refresh Tokens for Web Services** field is now visible.



4. Click to enable the check box next to **Enable Refresh Tokens for Web Services**. The menu expands and the **Maximum Token Refreshes Allowed** field is now visible.



5. Enter a numeral in the box next to **Maximum Token Refreshes Allowed**.
6. Scroll to the bottom of the page and click **Save**.
7. Authenticate with REST. You should receive both an access_token and a refresh_token.
8. Use the access token until it expires.
9. When the access token expires, POST to the same endpoint for authentication ("oauth2/token") with the body containing the following:

```
grant_type = "refresh_token"
Set refresh_token = <YOUR REFRESH TOKEN>
```

10. You should receive a refresh_token and a new access_token.

Example

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
$uri = "https:// < yoursecretserverinstance >"
$api = "$uri/api/v1"
function Authenticate {
    $args= @{
        username = "username"
        password = "password"
        grant_type = "password"
    }
    echo "----------------"
    echo "--Authenticate--"
    echo "----------------"
    $response = Invoke-RestMethod "$uri/oauth2/token" -Method Post -Body $args -
ContentType "application/json"
    $global:token = $response.access_token
    $global:refreshToken = $response.refresh_token
    $global:headers = New-Object "System.Collections.Generic.Dictionary[[String],
[String]]"
    $global:headers.Add("Authorization", "Bearer $token")
}
function Refresh {
    $args= @{
        grant_type = "refresh_token"
        refresh_token = $refreshToken
    }
    echo "---------------"
    echo "----Refresh----"
    echo "---------------"
    echo "--Sending Refresh Token"
    echo $refreshToken
    $response = Invoke-RestMethod "$uri/oauth2/token" -Method Post -Body $args -
ContentType "application/json"
    if($response.access_token){
        $global:token = $response.access_token
       $global:refreshToken = $response.refresh_token
        $global:headers = New-Object "System.Collections.Generic.Dictionary[[String],
[String]]"
        $global:headers.Add("Authorization", "Bearer $token")
    }
}
```

OpenID Connect

OpenID Connect (OIDC) in Secret Server is an identity protocol built on top of the OAuth 2.0 framework, designed to facilitate secure and streamlined authentication processes. By integrating with external OpenID Connect providers such as Azure AD, ADFS, Auth0, or Okta, Secret Server enables single sign-on (SSO) capabilities, allowing users to authenticate using their existing credentials from these providers. This integration not only enhances security by leveraging robust authentication mechanisms but also simplifies user management and access control. Administrators can configure Secret Server to delegate authentication to these external providers, ensuring a seamless and secure login experience for users across various applications and services.

OpenID Connect Integration

This applies to Secret Server Version 10.7 SP2+.

Introduction

OpenID Connect

OpenID Connect is an industry-standard single-sign-on (SSO) protocol. An identity provider implementing OpenID Connect can be used as an identity source for Secret Server, allowing users to log in with external credentials.

OpenID Connect Support in Secret Server

Secret Server

Secret Server implements OpenID Connect authorization code flow, allowing any standards-compliant provider to be used as an identity source. To use OpenID Connect, a Secret Server administrator must configure the login integration. Additionally, user accounts must be created in Secret Server that correspond to the external accounts.

Prerequisites

General

- Secret Server version 10.7 SP2+
- An OpenID-Connect-compatible identity provider, such as Delinea One, Azure AD, Auth0, or Okta.

Permissions

- To configure the login integration, a Secret Server user must have Administer OpenID Connect permissions. Secret Server
- To add user accounts that can be used with OpenID Connect, a Secret Server user must have administer user permissions.

Configuration

Task One: Acquire and Configure an OpenID Connect Provider

1. Follow your OpenID Connect provider's setup instructions for configuring a new identity client.
2. Gather the configuration data from the provider. To configure an OpenID Connect login provider, the provider must supply these:
   - **Provider URL**: An HTTPS URL, acting as an authentication endpoint. Secret Server expects the provider URL to be the "issuer" URL of the provider. For example, if the OpenID Connect configuration for the provider is accessible at

     `https://example.com/oidc/endpoint/default/`,

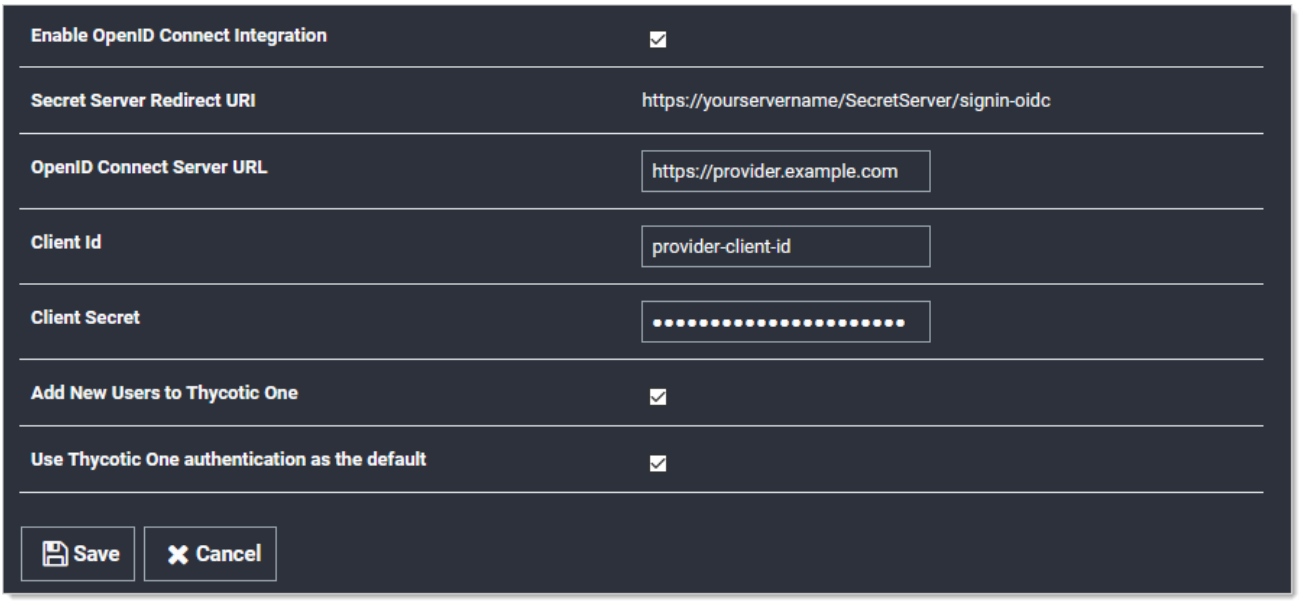
     then the URL used in the Secret Server configuration should be

     `https://example.com`.
   - **Client ID**: The ID portion of the credentials used to interact with the provider.
   - **Client Secret**: The password portion of the credentials used to interact with the provider.

   The process for determining this information varies by provider and you can usually find it by following the provider-specific documentation for configuring an OpenID Connect client.
3. Configure the provider with the Secret Server Redirect URI, as shown below, when configuring OpenID Connect integration in Secret Server. It is usually `https://YourWebServer/SecretServer/signin-oidc`. The Secret Server Redirect URI value must be added to the provider's configuration, so that the Secret Server instance can perform the authentication handshake. This process varies by provider, but it is usually known as a post-login redirect URI or callback URI.

Task Two: Configure Secret Server

1. Locate the Secret Server OpenID Connect configuration at **Admin > Configuration > Login**:



2. Click to select the **Enable OpenID Connect Integration** check box.
3. Fill in the other values using the what you gathered earlier.

> The two Thycotic One options are not relevant unless you use Thycotic One as your OpenID Connect provider. They have no effect for other providers.

4. Click the **Save** button.

Task Three: Matching External Accounts to Secret Server Users

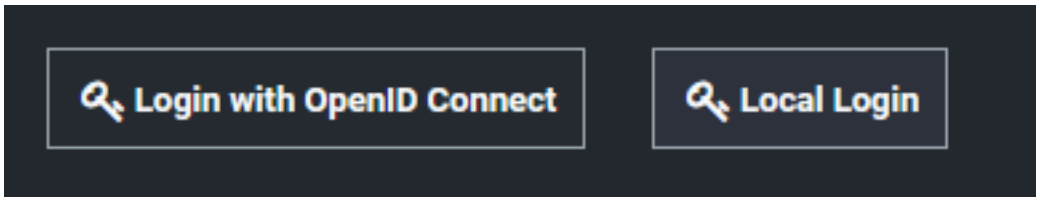
Ensure a matching user already exists in Secret Server, which is required when logging on an OpenID Connect account. OpenID provides for no user list synchronization or on-the-fly account creation. Users are matched according to the claims provided in their authentication ticket. Matching occurs using the following criteria:

- If the name identifier value matches an active user that has already logged in with OpenID Connect, then this user will be logged on.
- If not, then if the email or UPN values match an existing, unique, active Secret Server user, then this user will be logged on.

Otherwise, the login attempt will fail, and information about the failure will be added to the system log.

Task Four: Logging on with OpenID Connect

Finally, confirm the configuration by logging on with OpenID Connect. Once Open ID Connect has been configured, the Secret Server login page will have a new **Login with OpenID Connect** button:



Clicking this button initiates the OpenID Connect log on process with the external provider. Depending on provider settings, you may be asked to approve the login request or grant access to specific profile info. Once you have approved the request and logged in to the external provider, you will be redirected back to Secret Server and logged in as the corresponding local user.

Thycotic One and Secret Server

Overview

Thycotic One is a legacy single-sign-on provider for Delinea applications. With Thycotic One, one user account can be granted access to multiple Delinea products, such as Secret Server, Privilege Manager, DevOps Secrets Vault, and Account Lifecycle Manager. Thycotic One enables login integration using the OpenID Connect protocol, an industry standard single-sign-on method.

This article describes the Thycotic One configuration options available in Secret Server.

Cloud versus On-Premise

For Secret Server Cloud, Thycotic One is the default identity provider. When you set up the cloud instance, it will already be configured and ready to use Thycotic One. The initial admin user will log in with their Thycotic One account, and optionally, all newly created Secret Server accounts can be synchronized with Thycotic One, so they can log in that way as well.

For the on-premise version of Secret Server, Thycotic One integration is off by default, but it is supported. You can turn on Thycotic One integration and configure it. For example you might want to share an identity provider between your on-premise instance, and one or more other cloud products.

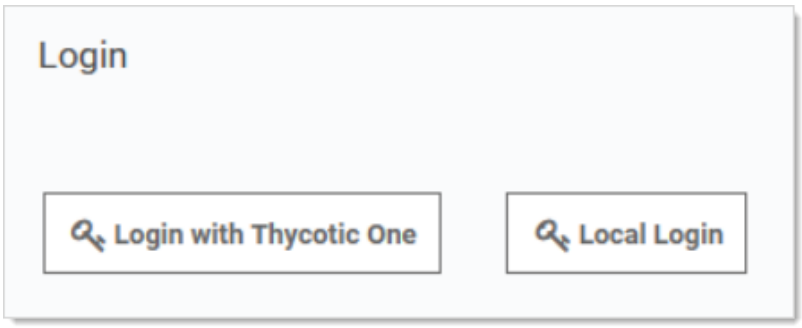
Procedures

Logging in with Thycotic One

When Thycotic One integration is turned on, all Secret Server users can log in either with their local passwords or with Thycotic One. All Secret Server permissions and configuration will apply to that user regardless of how they logged in.

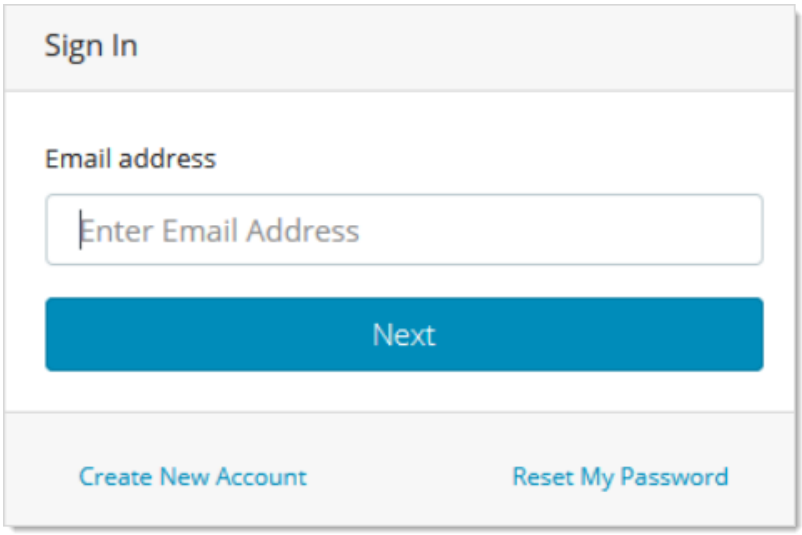
However, the local username and password and the Thycotic One username and password are not necessarily the same thing. In Thycotic One, you'll log in with your email address rather than your username, and the password you use may very well be different from the Secret Server password.

You'll see this on the login screen:



Clicking **Local Login** will bypass Thycotic One and allow the user to log in with their local Secret Server password. Clicking **Login with Thycotic One** will redirect the user to Thycotic One to authenticate. Once that is successfully done, the user will be redirected back to Secret Server.

After clicking **Login with Thycotic One**, users will type their email address and password:



And then be redirected back to their dashboard in Secret Server.

Configuring Thycotic One

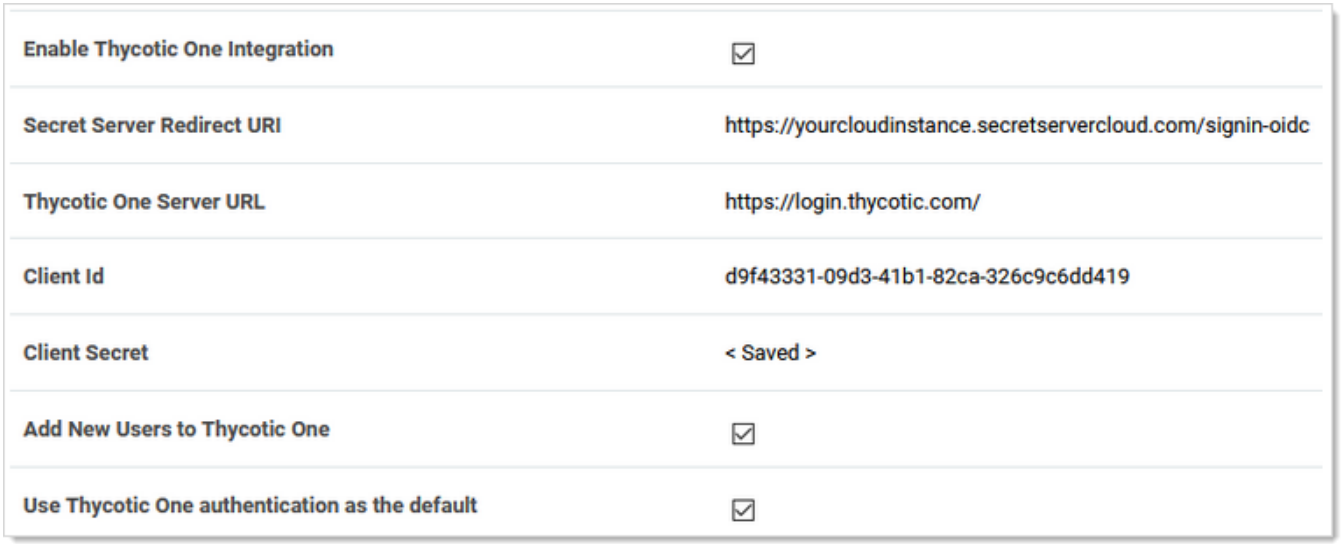
Thycotic One integration is configured on the **Admin** > **Configuration** page, under the **Login** tab. You can view the configuration there:

The **Sync Now** button provides a way for you to trigger a synchronization of your Secret Server accounts with Thycotic One. In most cases, you will not need to use this, as synchronization will happen on a schedule or whenever a relevant event happens, such as enabling a user or performing an Active Directory synchronization. Only active user accounts with email addresses will be synchronized.

Click **Edit** at the bottom of the page to change the configuration. The available options are slightly different between the cloud and on-premise versions of Secret Server.

Secret Server Cloud

When editing the options in Secret Server Cloud, you'll see something like this:



Here are the available options:

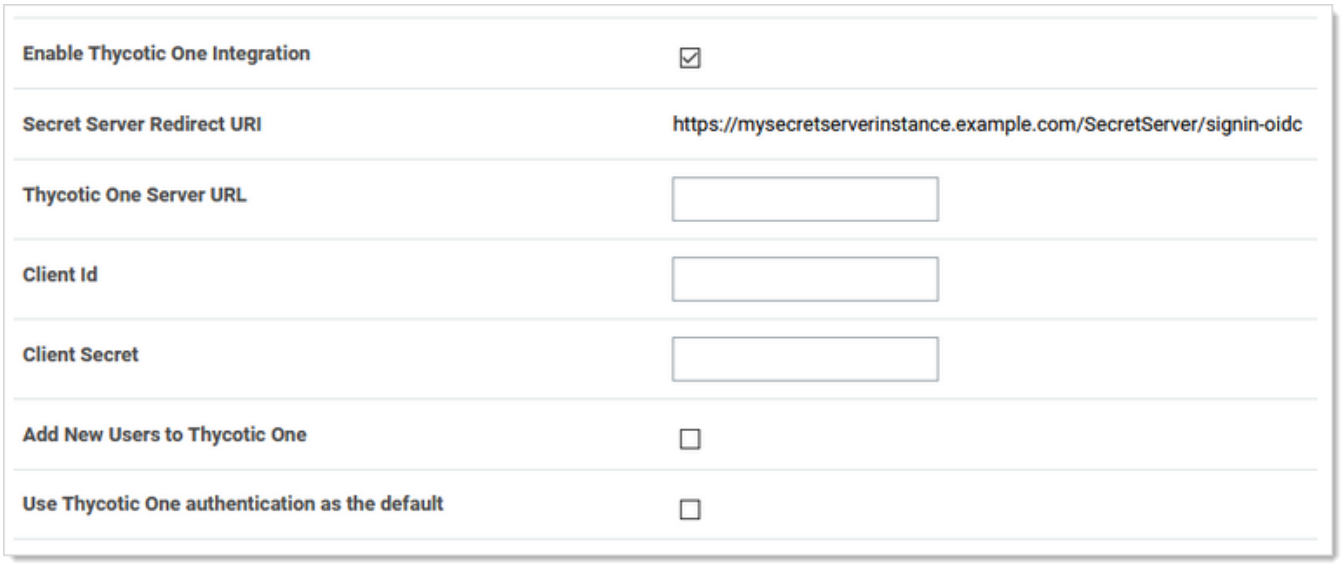
- **Enable Thycotic One Integration:** Turn on to enable Thycotic One functionality. Turn off to completely disable Thycotic One logins and synchronization. Make sure you have an admin account with a working local password.
- **Secret Server Redirect URI:** For informational purposes, this shows the page address to which you are redirected after you have logged in with Thycotic One.

- **Thycotic One Server URL:** The Thycotic One server you have connected to. There is one separate Thycotic One instance in each Secret Server Cloud region.
- **Client ID:** The client ID portion of the Thycotic One server credentials.
- **Client Secret:** Not shown, the client password portion of the credentials.
- **Add New Users to Thycotic One:** When checked, Secret Server accounts will be synchronized with Thycotic One. Adding a user will send them a welcome email, where they can set up their Thycotic One account password and log into Secret Server. When unchecked, users will not be synchronized and no email will be sent. New users will not be able to log in with Thycotic One, unless you click **Sync Now** on the **Admin** > **Configuration** > **Login** page, which will synchronize all active users.
- **Use Thycotic One authentication as the default:** When checked, Thycotic One authentication is used for the REST and SOAP APIs and mobile apps. Users who have logged in with Thycotic One use their Thycotic One account passwords for those activities, rather than their local Secret Server account passwords. When unchecked, they will use their local Secret Server account passwords for those activities.

In Cloud, the server URL, client ID, and client secret cannot be edited—they are set up for you when the instance is provisioned and cannot be changed.

Secret Server On-Premise

When editing the options in Secret Server on-premise, you'll see something like this:

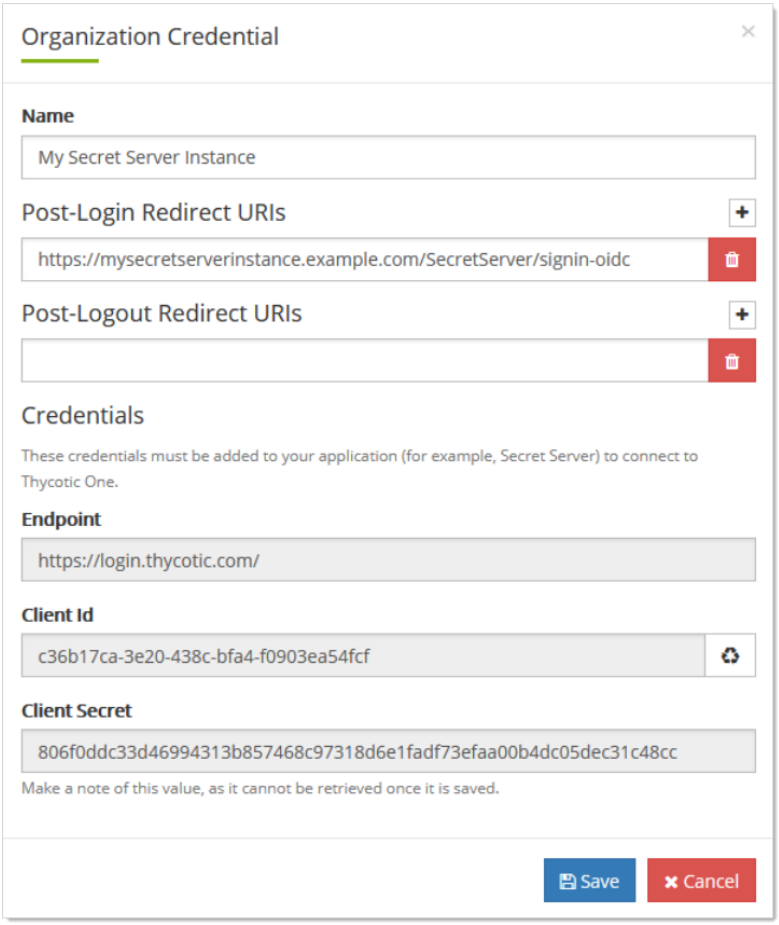


Unlike in Cloud, the server URL, client ID, and client secret can be edited in an on-premise instance. You can generate Thycotic One credentials using Delinea's cloud management portal, Cloud Manager. Otherwise, the configuration options behave the same as in Cloud.

Generating a Thycotic One Credential

To generate a credential for use in an on-premise Secret Server instance, follow the steps below:

1. From Cloud Manager, choose a Thycotic One region under Other Login Options.
2. Log into Thycotic One as a user that will be managing your organization's credentials. Create an account if you have not yet done so.
3. Go to Cloud Manager.
4. Click **Sign In**. You are redirected to our tech support portal login.
5. Click the button for the Thycotic One region you chose. Since you are already logged in to Thycotic One, this will redirect you back to Cloud Manager.
6. Next, choose a team: In the menu, go to **Manage** > **Teams**. You may already have one if you have an existing cloud product. If not, create one. Each team can handle multiple Thycotic One credentials.
7. Having selected your team, go to **Organizations**. Again, if you already have an organization, you can use it; if not, you can create one. An organization provides a way to manage the global login policies for all users.
8. Go to **Credentials**. Click **Add**. An Organization Credential dialog box appears:



9. The available fields are as follows:

- **Name:** A description of the application using this credential, for informational purposes.
- **Post-Login Redirect URIs:** A list of valid URIs that will be allowed to authenticate with this credential. The value of "Secret Server Redirect URI" from your on-premise instance should go here. If users access your instance with more than one URI, you may want to add all of them here by clicking the **+** button to create additional fields. Unless an application supplies a URI that is an exact match to one of these, Thycotic One will not complete the authentication.
- **Post-Logout Redirect URIs:** Secret Server does not support this feature, so this may be left blank.
- **Credentials:** The fields in this area contain the values you need to put into the Thycotic One configuration in SS. Copy and paste them into the corresponding fields.

10. Once you capture all the values, click **Save**, and then save the configuration in Secret Server as well. Your instance is now fully integrated with Thycotic One. If you selected the synchronization option, Secret Server will immediately sync your active users with Thycotic One, and they'll receive welcome emails describing how to continue the process.

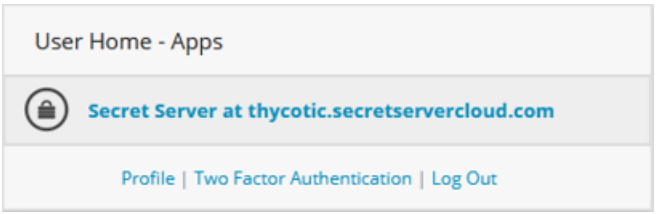
Enabling Two-Factor Authentication in Thycotic One

When two-factor authentication is enabled, Thycotic One presents a two-factor challenge to the user logging in. The Thycotic One two-factor authentication supplements and does not replace any other two-factor authentication methods used by a client application such as Secret Server. Thycotic One supports two-factor authentication using TOTP or SMS. You can have only one two-factor authentication method active at any time. We recommend using TOTP over SMS whenever possible for better security.

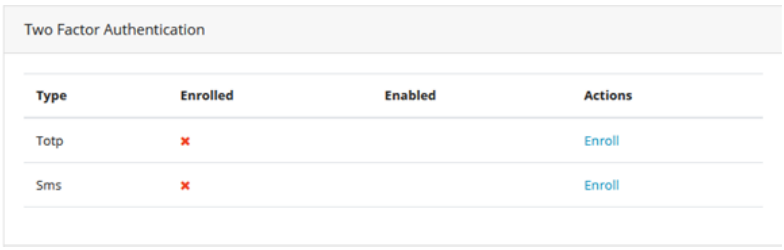
TOTP Two-Factor Authentication

To use TOTP two-factor authentication with Thycotic One, you must first have a mobile device with an installed TOTP application such as Google Authenticator, Authy, or Microsoft Authenticator. When you have the app installed, follow the steps below.

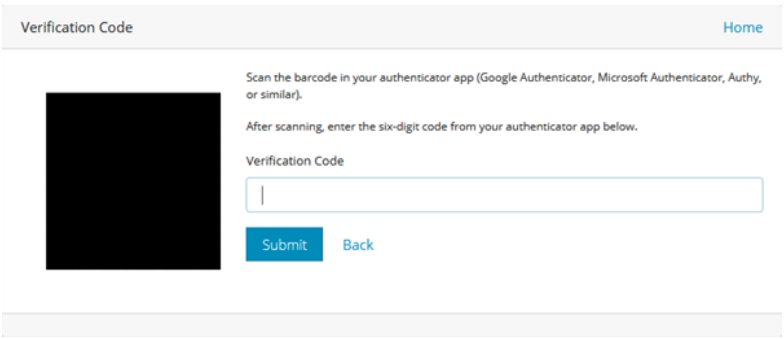
1. Log intoThycotic One and on the account homepage in the **User Home - Apps** dialog, click **Two-Factor Authentication**.



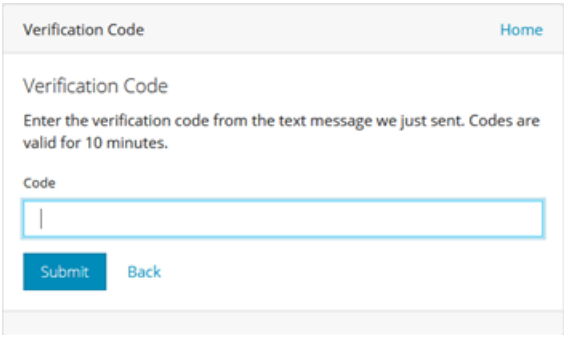
2. Choose **TOTP** and click **Enroll**.



Thycotic One displays a barcode (redacted in the example shown).



3. Using the TOTP app on your mobile device, scan the barcode. You will receive multiple six-digit codes.
4. In the Verification Code field, enter one of the six-digit codes (a new code is generated every 30 seconds).

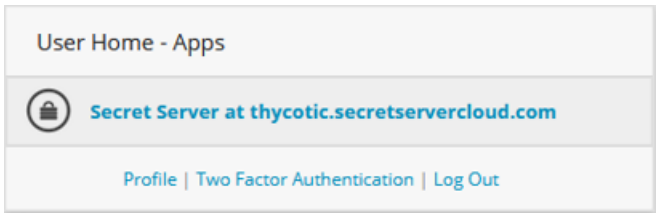


When you have correctly entered and submitted a six-digit code, the setup of TOTP two-factor authentication is complete. From this point forward, each time you attempt to log in you will receive a text message on your mobile device with a code that you must enter to complete the login process.

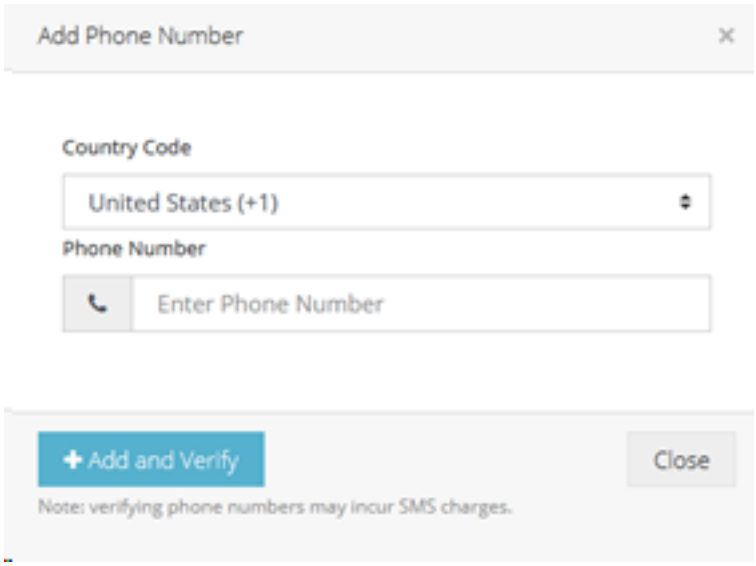
SMS Two-Factor Authentication

To use SMS two-factor authentication with Thycotic One, you must first provide and verify a mobile phone number.

1. To provide a mobile phone number, log into Thycotic One, and on the account homepage in the **User Home - Apps** dialog, click **Profile**.



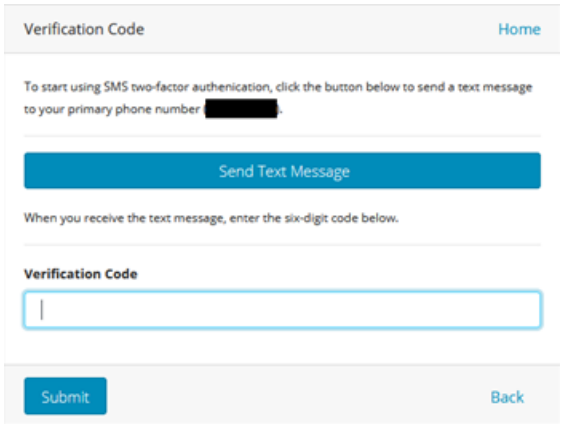
2. Click **Add Phone** and enter the country code and phone number of a mobile phone that accepts text messages.



3. Click **+Add** and **Verify**.

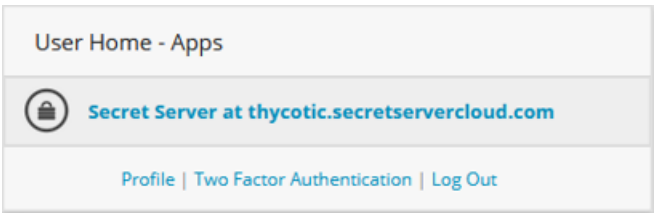
   Delinea One sends a text message to the phone number, with a code.

4. Enter the code and click **Submit** in the Verification Code dialog.

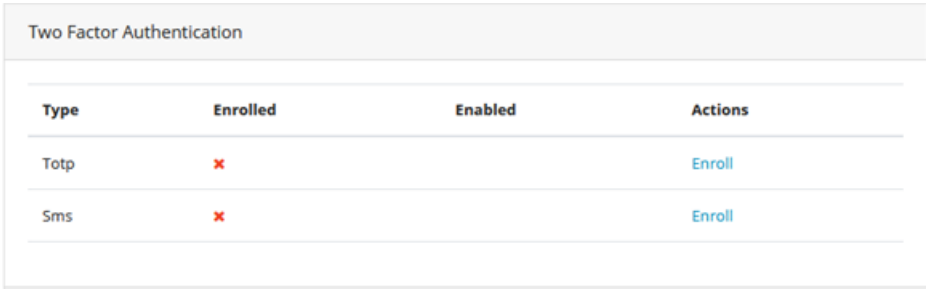


   The phone number now appears as Verified on your profile page.

5. 1. On the Thycotic One account homepage in the **User Home - Apps** dialog, click **Two-Factor Authentication**.

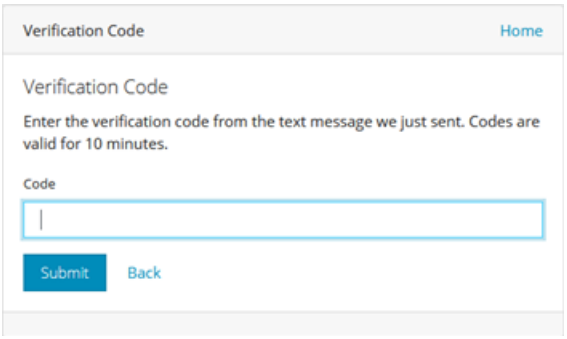


6. Choose **SMS** and click **Enroll**.



   Delinea One sends a text message to your phone with a six-digit code.

7. Enter the six-digit code into the box provided.



When you have correctly entered and submitted the six-digit code, the setup of SMS two-factor authentication is complete. From this point forward, each time you attempt to log in you will receive a text message on your mobile device with a code that you must enter to complete the login process.

SAML

SAML (Security Assertion Markup Language) in Secret Server enables secure, single sign-on (SSO) authentication by allowing Secret Server to act as a SAML Service Provider (SP) that communicates with any configured SAML Identity Provider (IDP). This integration facilitates seamless user authentication across various applications and services, leveraging existing identity management systems such as Okta, OneLogin, Azure AD, and Microsoft ADFS.

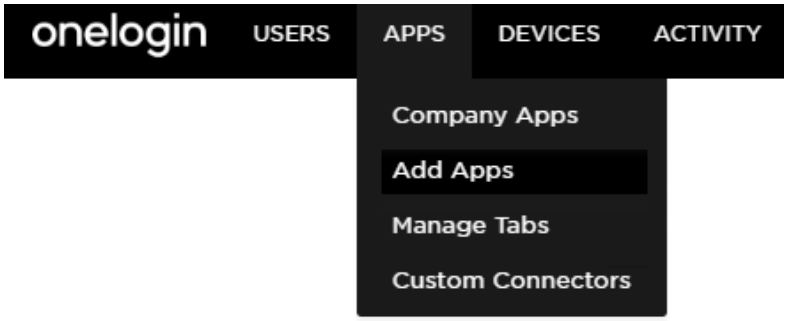
By using SAML, organizations can streamline the authentication process, enhance security through centralized identity management, and provide a consistent multi-factor authentication (MFA) strategy across their environment. Configuring SAML in Secret Server involves setting up the SAML Service Provider, importing the necessary certificates, and configuring the IDP to ensure secure and efficient authentication workflows.

Configuring SAML OneLogin

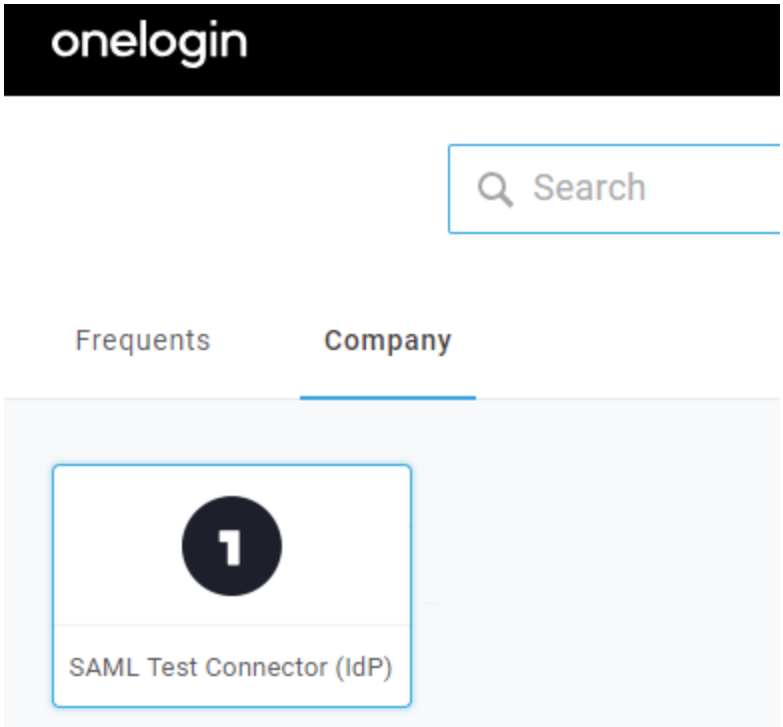
To access Secret Server using OneLogin for SAML, follow the steps below for OneLogin, then follow the steps for Secret Server.

Step One: OneLogin

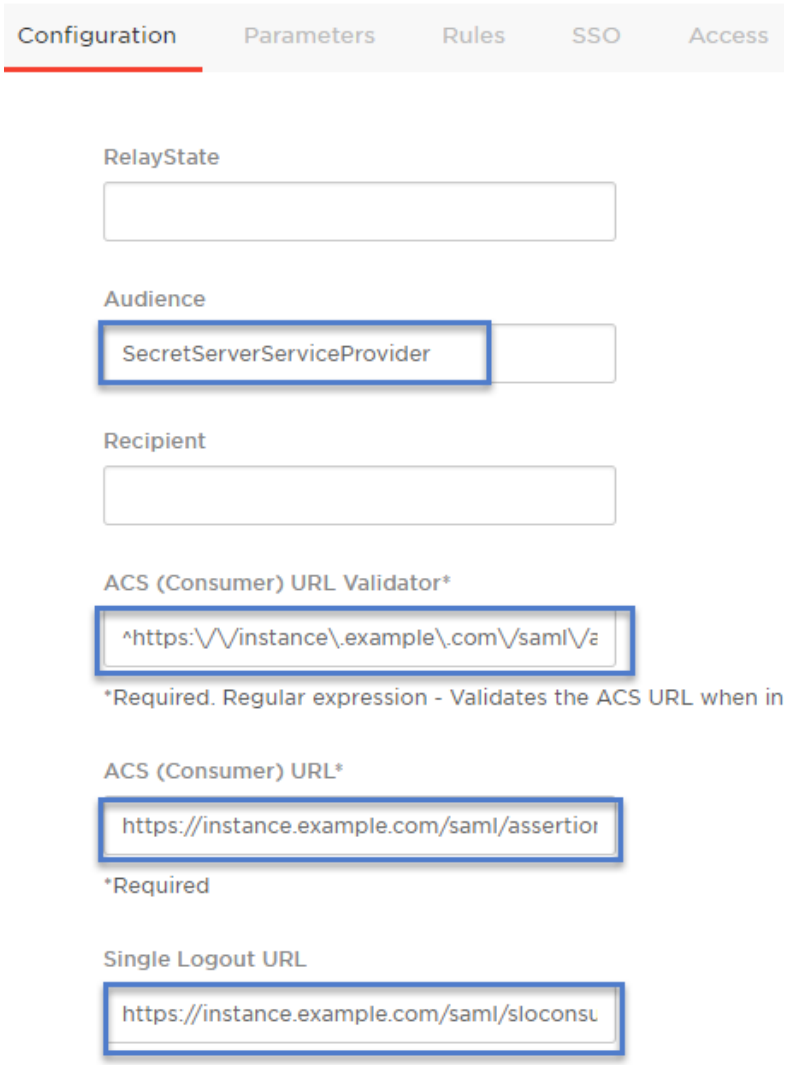
1. Navigate to your OneLogin instance and log in as an administrator.
2. Select **Administration > Apps > Add Apps**.



3. Search for **SAML Test Connector (IDP)** and select it, then click **Save**.



4. Click on the **Configuration** tab and fill out the details as described below:



- **RelayState** can be left blank.
- **Audience** is the name of the Service Provider configured in Secret Server (for instance "SecretServerServiceProvider").
- **Recipient** can be left blank.
- **ACS (Consumer) URL Validator** a required field that needs to be a valid RegEx of the ACS (Consumer) URL.

  Modify the text in the example below according to the URL string of your Secret Server instance:

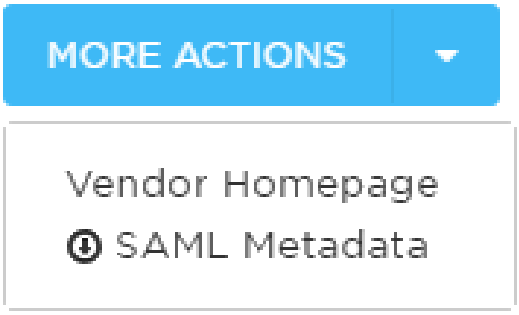
  `https:\/\/instance\.example\.com\/saml\/assertionconsumerservice\.aspx$`
- **ACS (Consumer) URL** like the step above, but no longer in RegEx format.

  Modify the text in the example below according to the URL of your Secret Server:

  `https://instance.example.com/saml/assertionconsumerservice.aspx`

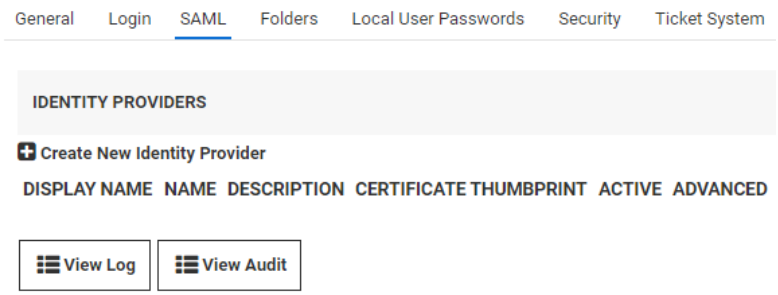
- **Single Logout URL** the Secret Server URL for SLO (Single Logout):
  `https://instance.example.com/saml/sloservice.aspx`

5. Click **Save** when done.
6. Click **More Actions** and **SAML Metadata** to download the metadata for OneLogin.

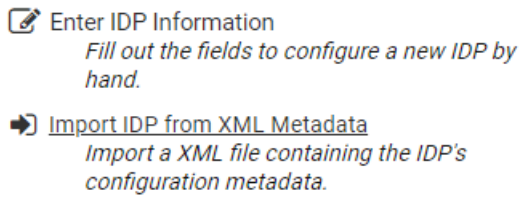


Step Two: Secret Server

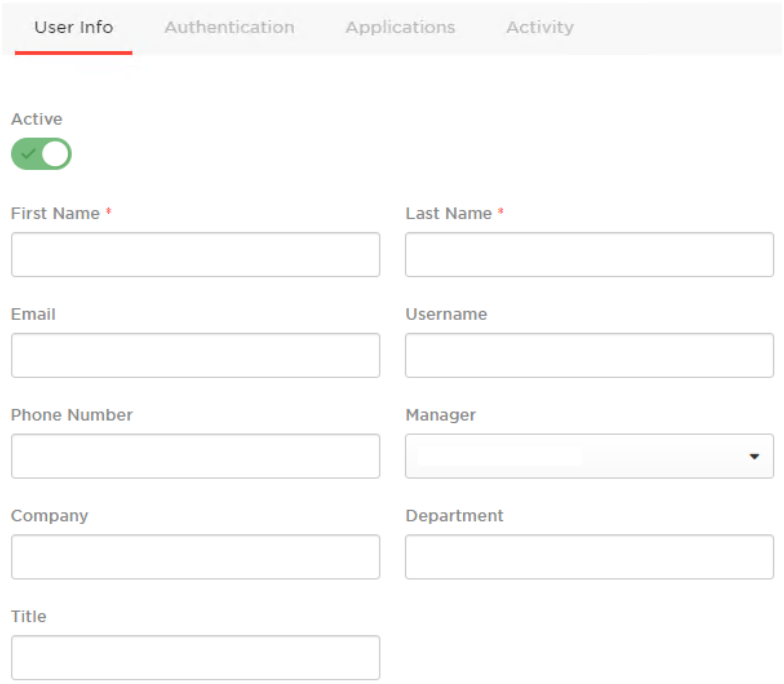
1. Log into your Secret Server instance, then go to **Admin** > **Configuration** > **SAML** tab and click **Create New Identity Provider**.



2. Click **Import IDP from XML Metadata** and select the OneLogin metadata you saved previously. If you don't see the file, you may need to change the metadata filetype to .xml



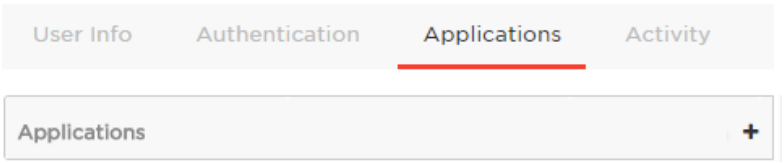
3. To add users to OneLogin, navigate to OneLogin and log in as an administrator once more, then click **Administration** > **Users** > **New User**.



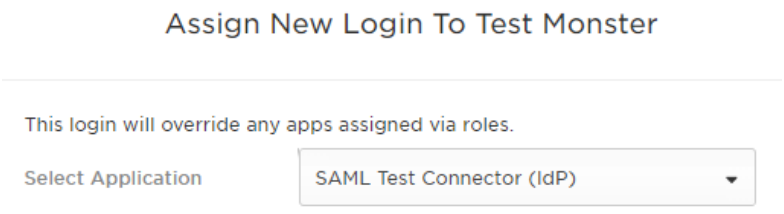
4. Fill out the required information and click **Save** when finished.

> If you are using a Secret Server local account or Secret Server Cloud, the username will be in email format and it must be identical on OneLogin and Secret Server. For an Active Directory account, it should be the samAccountName.

5. Click on the **Applications** tab, then click the plus sign (+).

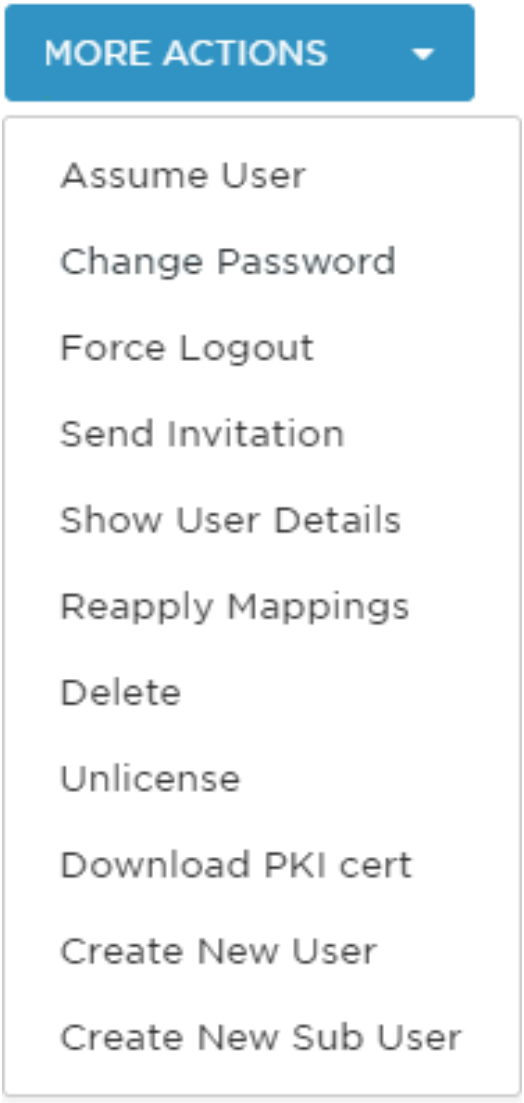


6. Select **SAML Test Connector (IDP)**, then click **Continue**.



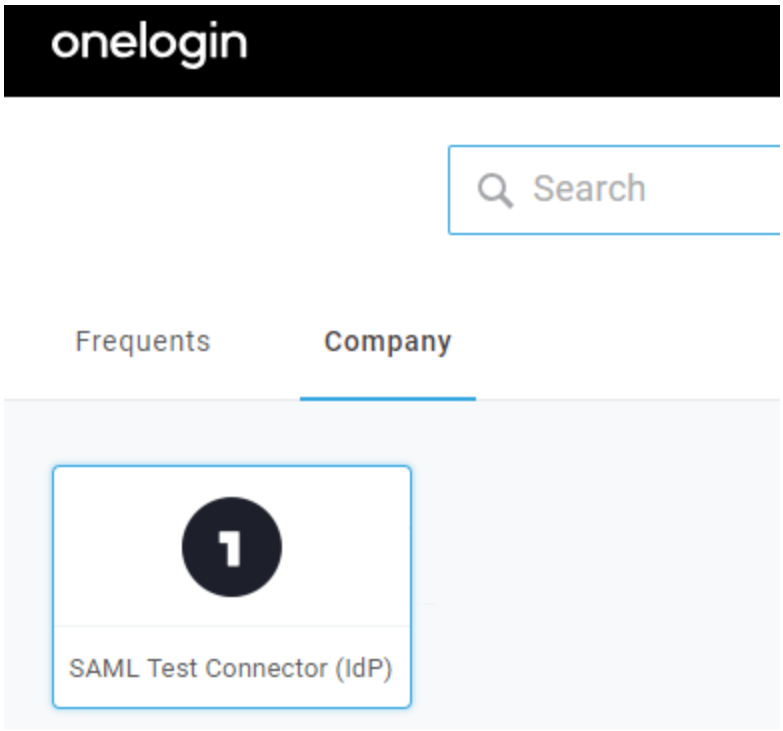
7. Enter the user's Secret Server username (email format) then click **Save**.

8. Mouse over **More Actions** and click **Change Password** to give the user a login password.



9. In another browser or in incognito mode, log into your OneLogin instance as the user you just created. If prompted to add OneLogin to your browser, click **Skip**.

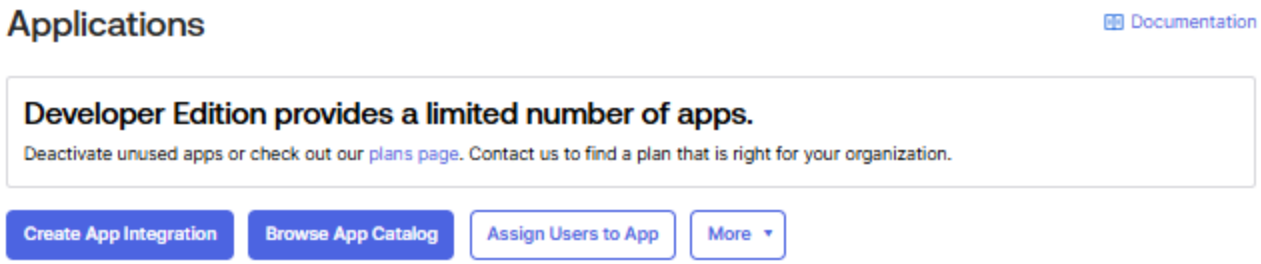
10. You should see the **SAML Test Connector (IDP)**. Click on it to authenticate into Secret Server using the SAML workflow.



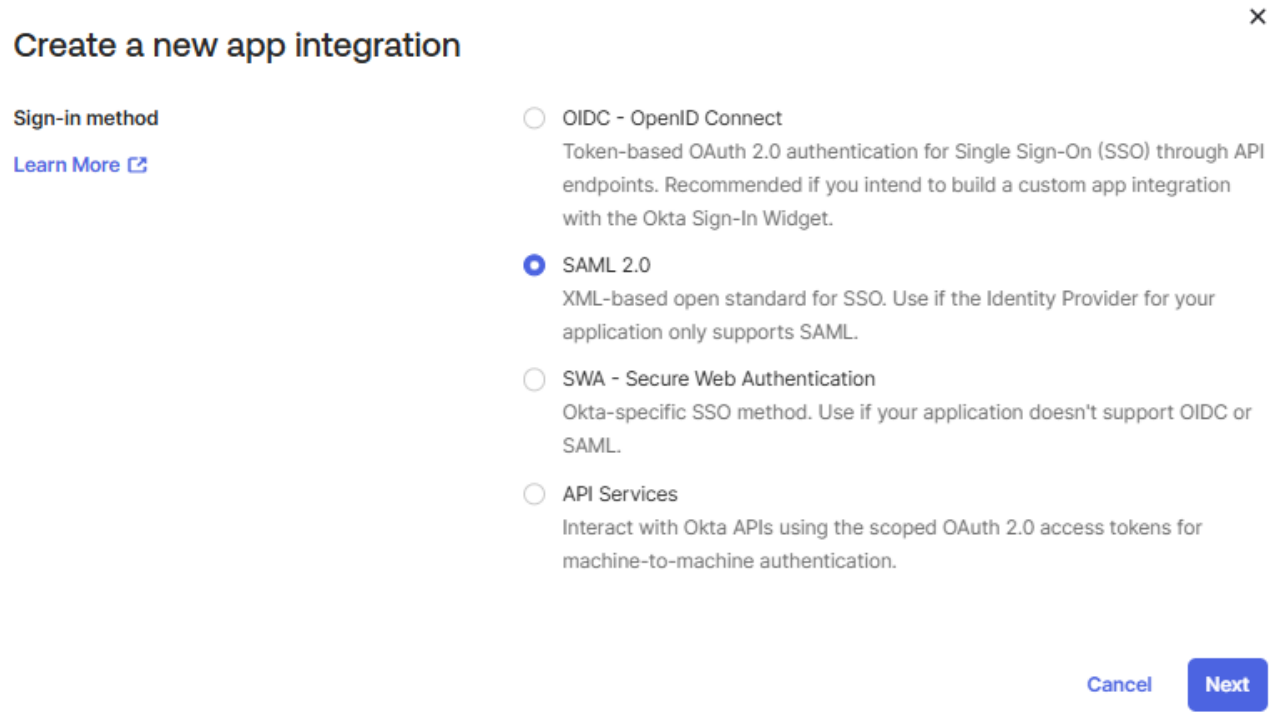
Configuring SAML Okta

Creating the Application Integration

1. Log into your Okta instance using an administrative account.
2. From the left-hand menu, click **Applications > Applications**.
3. Select **Create App Integration**:



4. In the **Create a new app integration** pop-up, select **SAML 2.0**, and click **Next**:

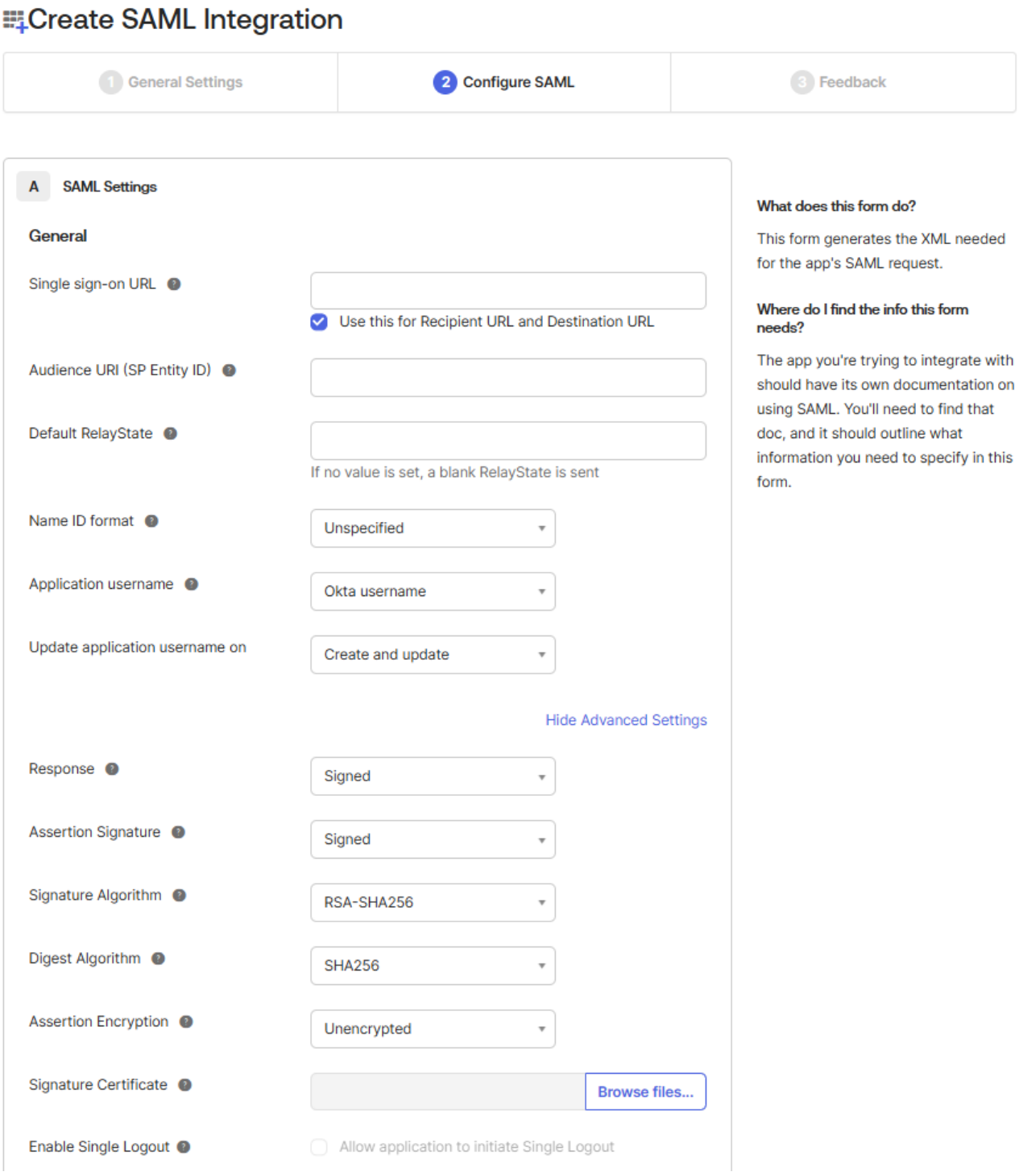


5. The **Create SAML Integration** page loads.
6. In the **General Settings** tab, type the preferred name (for example, SecretServer) in the **App name** field, and click **Next**.

   The App name field is the only required input on this screen.

7. In the **Configure SAML** tab, in the **A SAML Settings** section, under **General** fill in the following mandatory fields:
   - **Single sign-on URL**: your [SecretServerInstanceName] URL.

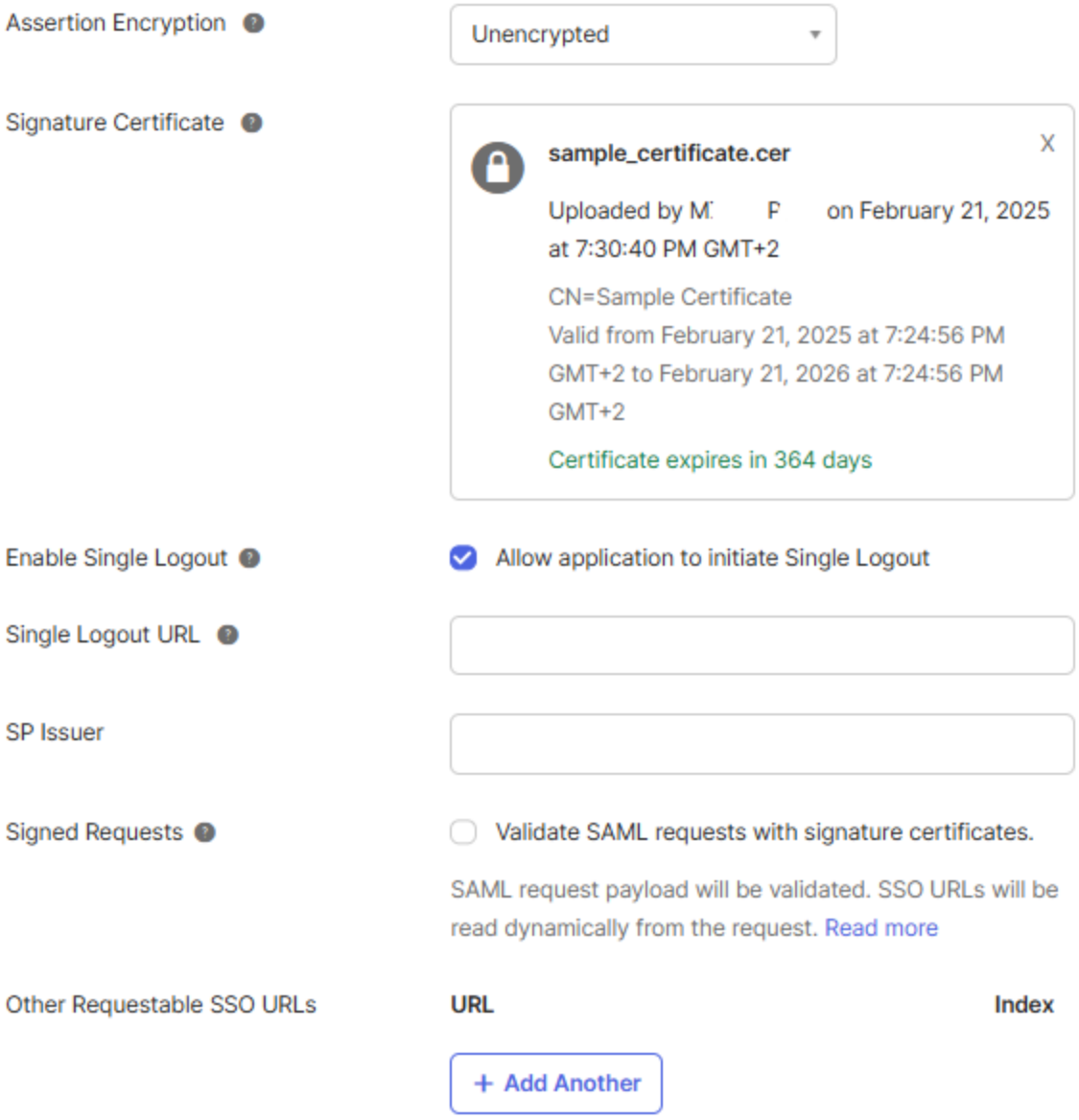
     For example: https://[YourSecretServerInstance.com]/saml/assertionconsumerservice.aspx
   - Make sure the **Use this for Recipient URL and Destination URL** checkbox is selected.
   - **Audience URI (SP Entity ID)**: SecretServerServiceProvider.
   - **Default RelayState**: leave the field blank.
   - **Name ID format**: select **Unspecified** from the dropdown list.
   - **Application username**: select **Okta username** from the dropdown list.
8. Click **Show Advanced Settings** . A plethora of settings will load.
9. Look for the **Signature Certificate** option:



10. Click **Browse files...** to locate your public key `.cer` file.

> The file corresponds to the .pfx file uploaded to your Secret Server instance. See "Configuring SAML Single Sign-on" on page 422 for details on configuring SAML SSO and uploading Certificates.

11. Once chosen, the `.cer` file displays in the **Signature Certificate** field.
12. Select the **Enable Single Logout** checkbox below the **Signature Certificate** field:



13. Configure the Single Logout (SLO) Settings as follows:
    - **Single Logout URL**: your [SecretServerInstanceName] followed by the URL string: /saml/sloservice.aspx.

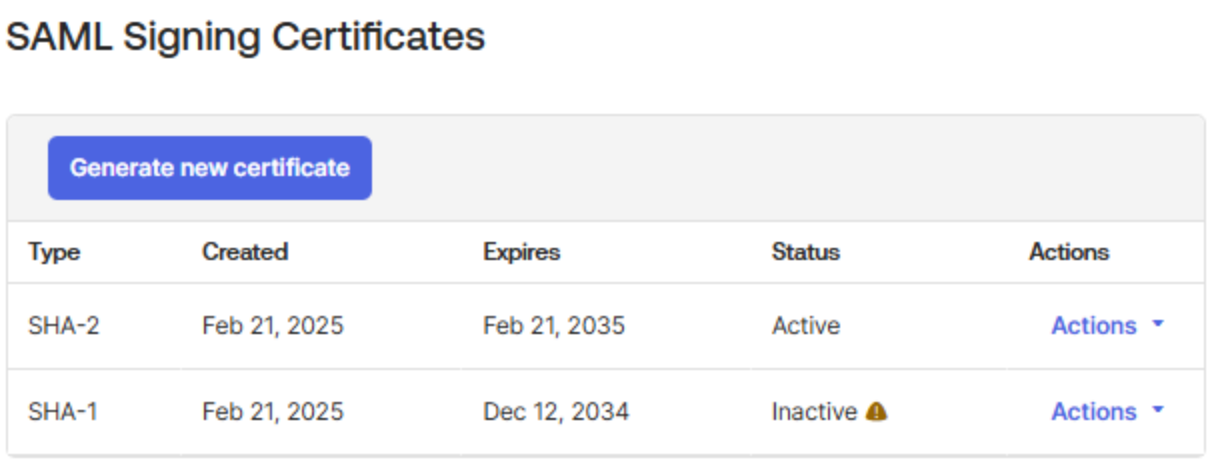
      Example: https://[YourSecretServerInstance.com]/saml/sloservice.aspx

- **SP Issuer**: SecretServerServiceProvider

14. Click **Next**.
15. (Optional) In the **Feedback** page, provide feedback.
16. Select **Finish**. The "SecretServer" integration has now been created.

Downloading the IDP Metadata File

1. In your Okta instance access **Applications > Applications**, and click on the **Active** option under **Status**.
2. Select the SAML app you previously created "SecretServer". The app opens on the **Assignments** tab.
3. Select the **Sign On** tab.
4. Go to the **SAML Signing Certificates** section and look for the certificate marked **Active** in the status column.
5. Select the **Actions** dropdown menu for that certificate and click **View IdP Metadata**:



6. Once the IdP metadata page loads displaying the XML file, right-click the page and select **Save As** or **Save Page As** (depending on the browser).
7. Save the XML file as `metadata.xml`. You will need this file to create the new SAML Identity Provider in Secret Server.

Configuring SAML in Secret Server for Okta

1. Log into Secret Server.
2. Access **Settings > Configuration > SAML > Identify Providers**.
3. Select the **Create New Identity Provider** button.
4. In the Identity Provider popup, select the **Import IDP from XML Metadata** option from the dropdown. The **Import file** option appears.
5. Click the **Change** link to import the Okta metadata XML file you saved earlier.
6. Click **OK**.

Verifying the Integration

Ensure that the integration works by testing the SAML login process. Users should be able to log into Secret Server using their Okta credentials without being prompted for additional login information. This setup will allow you to use Okta as the Identity Provider for SAML-based single sign-on in Secret Server, streamlining user authentication and enhancing security.

Configuring SAML Single Sign-on

This topic is for Secret Server v10.5 and later and assumes you have a running Identity Service Provider (IDP) with a signed certificate.

Secret Server does not support using SAML when Integrated Windows Authentication (IWA) is enabled.

This topic applies to Secret Server 10.5 and later.

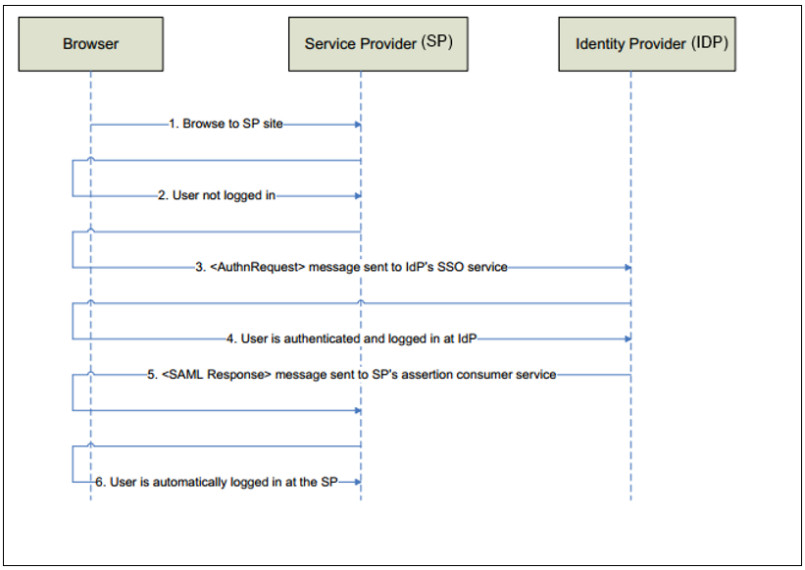
Sometimes SAML validates despite having an expired IDP Certificate. This is because the SAML certificate exchange is actually a key exchange—the SAML protocol uses the public/private keys contained within the certificate to secure the communication. Unlike a website's certificate, it is not the URL or metadata within the certificate that is used to help secure the connection, only the key exchange. If the certificate provided by the IDP has been renewed but still works without a change in Secret Server, it means the key is identical, and only the metadata, such as the expiration date, has changed. This is why it does not fail validation—the key is for validation and it has not changed. Had the key changed, Secret Server would not be able to decrypt the SAML messages, and would therefore fail. Delinea does not view this as a security problem and offers this note for clarification.

SAML Overview

Secret Server allows the use of SAML Identity Provider (IDP) authentication instead of the normal authentication process for single sign-on (SSO). To do this, Secret Server acts as a SAML Service Provider (SP) that can communicate with any configured SAML IDP.

In the diagram below, Secret Server acts as the service provider. Any configured SAML IDP can be used for this process and there are several well tested providers, including OKTA, OneLogin, Azure ADFS, and Microsoft ADFS.

Figure: Secret Server as a SAML Identity Provider



Prerequisites

Licensing and Version

Secret Server Professional Edition or higher, upgraded to version 10.5 or later. To install a new SAML license, go to **Admin > Licenses > Install New License**.

.NET Framework 4.6.2+

To use SAML 2.0, you must install .NET Framework 4.6.2 or higher on your Web server. This allows Secret Server to use Microsoft's "next generation" CryptoNG API for signing SAML requests, instead of being limited to the much older CryptoAPI. This is often necessary to use modern SSL certificates and is strongly recommended as a security best practice.

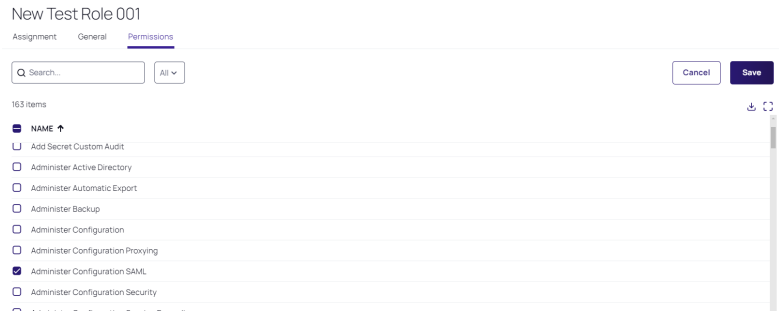
To download and install the latest version of .NET Framework: See Microsoft .NET Framework 4.8 offline Installer for Windows for the latest version as of when this topic was written. If you have already installed Secret Server on the same Web server, you have already done this.

Administer Configuration SAML Role Permission

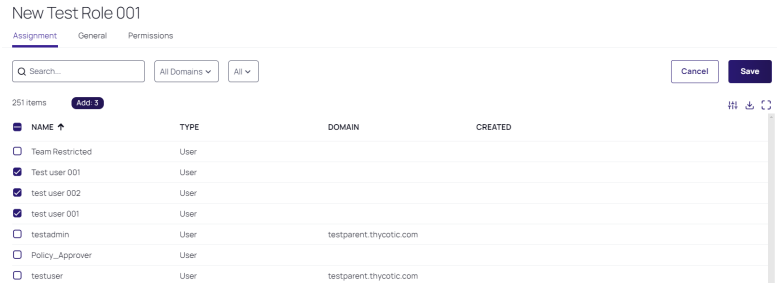
The "Administer Configuration SAML" role permission is required to configure SAML settings (no specific permission is required to access Secret Server via SAML). To grant a user this permission from an administrator account:

1. Go to **Admin > Roles**. The Roles page appears.
2. Click the **Create Role** button. The Create Role window appears:

3. Type the name, such as `SAML`, in the **Name** text box.
4. Click to select the **Enabled** check box.
5. Click **Create Role**. The role is created and the role page is now opened.
6. Under the **Permissions** tab, click the **Edit** button.
7. Select **All** next to the search box, select **Administer Configuration SAML** from the list, and click **Save**.



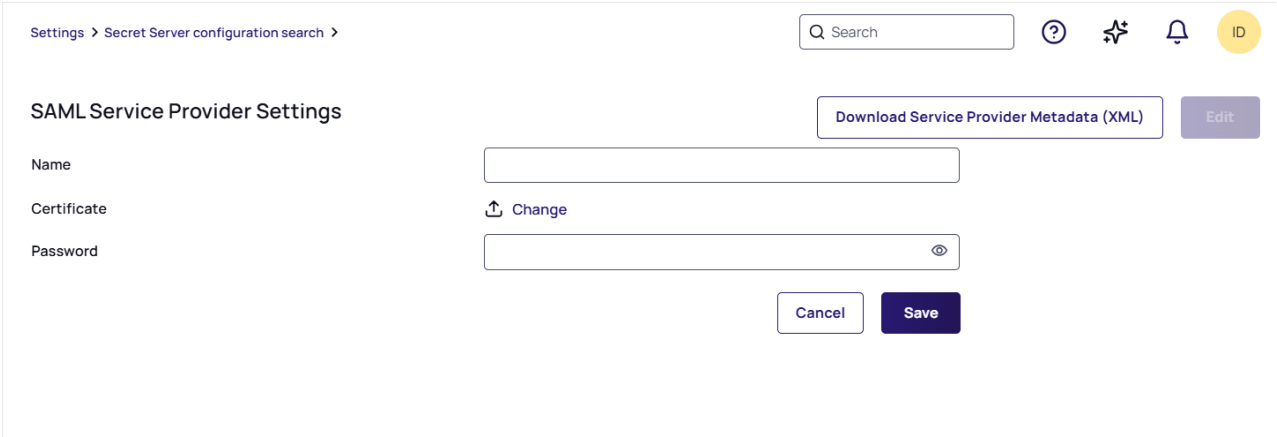
8. Under the **Assignments** tab, click **Edit**, select **All** next to the search box, select the users that you would like to assign to the role, and click **Save**.



Setting up Secret Server

1. Navigate to **Administration > Configuration Search**.
2. In the **SAML** section, click **General Settings**.
3. Click the **Edit** button.
4. Click to select the **SAML Enabled** check box.
5. Click the **Save** button.

6. Once you have SAML setup on our identity provider, go back to the **Settings** page and click **Service Provider Settings** in the **SAML**, section.
7. Click **Edit**.
8. Type a name for your Secret Server service provider, such as `SecretServerServiceProvider`, in the **Name** text box.
9. Click the **Change** button. The Upload Certificate popup appears:



What type of certificate can be used?

- The uploaded SAML certificate requires a `.pfx` file format.
- For on-premises instances: The uploaded certificate should match the one used for Secret Server's HTTPS configuration, **or** it can be created as a self-signed certificate using "Generating Self-Signed Certificates for Scripts" on page 1467.
- For Secret Server Cloud users: Generate your own certificate using the same PowerShell script.

> Run the referenced PowerShell script as an administrator on a machine with .NET 4.5 or above and replace the variables in the script as directed. Your certificate is created in the directory from which you run the script. The subject name on the certificate is irrelevant, though for on-premises instances it typically matches the URL of the instance.

10. Locate your certificate `.pfx` file and select it.
11. Click the **Open** button. The new certificate appears.
12. Type the access password for the private key of the certificate in the **Password** text box.
13. Click the **OK** button. The certificate is uploaded and tested, and the popup disappears. The certificate now appears in the SAML Service Provider Settings section.

> If you have an outdated version .NET Framework (earlier than 4.6.2), you may see an error recommending you upgrade to fix the error. Reload the certificate after you do so.

14. Click the **Save** button.
15. Click on **Download Service Provider Metadata (XML)** to download the `SecretServerSAMLMetatdata.xml` file. This file is needed when setting up SAML on your Identity Provider.
16. Set up your Identity Provider using the `SecretServerSAMLMetadata.xml` file from the previous step. Go to portal.azure.com and navigate to **Microsoft Entra ID**> **Enterprise Applications**.
17. Click the **Create New Identity Provider** link. An Identity Provider popup appears.
18. Click the **Import IDP from XML Metadata** button.
19. Navigate to your `SecretServerSAMLMetadata.xml` file and select it.
20. Click the **Open** button.

Setting up IDPs

IDP setup varies by provider. For Entra ID, go to "Setting up Entra ID for SAML" on page 514. For all other IDPs, go to the TDP Integration site for instructions for your provider.

> The username returned from the IDP to Secret Server within the SAML Response/Assertion's subject statement must match the desired format. The format of the username passed depends upon how the user was created within Secret Server. Users must be present in the User Management section to set up SAML.

> If AD Sync was used to create Secret Server users, the username returned from the IDP must match this format: `SecretServerUsername@ADsyncDomain` or`ADsyncDomain\SecretServerUsername`. If using SLO, ensure that the NameID is set correctly in the IDP as an outgoing claim for the Secret Server Service Provider. If a user has different sAMAccountName and userPrincipalName in Active Directory, custom rules in the IDP can be created. This requirement is applicable not only to Active Directory but also to OpenLDAP and Entra Integration. Regardless of the directory service used, the username or UPN values in the tbUser table must be utilized. If there is no UPN, then at least the domain\username from the tbUser table should be passed. And for the local users, it should simply pass the username value.

> When using Single Logout (SLO), ensure that the NameID is correctly set in the IDP as an outgoing claim for the Verify Privilege Vault Service Provider. In cases where a user has different sAMAccountName and userPrincipalName, custom rules in the IDP can be created to address this difference.

Lockout Workaround

Locked Out? Here's how you get around SSO. If during the configuration process for SAML you lock yourself (as an administrator or a user) out of Secret Server, you can log on Secret Server without using the SSO workflow by using this URL string:

`[YourSecretServerInstanceName]/login.aspx?preventautologin=true`

The role permission needed for this is "Bypass SAML Login," which admins have by default.

Generate a Self-Signed Certificate for Secret Server Using PowerShell

The script and steps required to generate a self-signed certificate can be found at "Generating Self-Signed Certificates for Scripts" on page 1467. This script can be used for the SAML configuration in a Secret Server instance. For additional information, refer to Configuring SAML Single Sign-On.

> Replace the variables in the script, as directed, and be sure to run the PowerShell script as an administrator on a machine with .NET 4.5 or above. The self-signed certificate will be created in the directory from which the script was run (e.g., C:\Users\Administrator).

> The subject name on the certificate is irrelevant. Although for Secret Server On-Premises, it typically matches the URL of the instance.

Smart Card Integration with Secret Server

Since Secret Server uses IIS to run the web application, we use the IIS function for smart card authentication: https://technet.microsoft.com/en-us/library/cc732116.aspx

We recommend enabling Integrated Windows Authentication in Secret Server (Under **Administration > Active Directory**) if the users are AD users. That way Secret Server will not prompt for credentials if the user is authenticated to AD. IIS will authenticate the users based on the smart card certificate, and Secret Server will pass the user through since they are logged in as an AD user.

To do this, there are two parts:

Configure Client Certificate Authentication in IIS (Learn more)

Configure Windows Authentication in IIS and Secret Server - "Configuring Integrated Windows Authentication" on page 383

1. Install the **Client Certificate Mapping Authentication** and **IIS Client Certificate Mapping Authentication** role services for IIS.
2. Reboot.
3. In IIS manager, highlight the server and click **Authentication**. Enable **Active Directory Client Certificate Authentication**.
4. In IIS manager, highlight the virtual folder for SecretServer and click **SSL Settings**. Check **Require SSL** and select Accept under **Client Certificates**.
5. Edit the file:

   ```
   %windir%\system32\inetsrv\config\applicationhost.config
   ```

   and add

```
<clientCertificateMappingAuthentication enabled="true" />” under the”
<windowsAuthentication enabled="false" />
```

6. Enable Integrated Windows Authentication in SecretServer. When the user goes to the website they get prompted for smart card credentials if there is one inserted. Otherwise it takes them to the login screen.

If you experience a performance issue with the first login, it may be related to the SSLAlwaysNegoClientCert property. More on this can be read here: "Changing IIS to Not Stop Worker Process in IIS 7.0 and Later" on page 238

That could affect performance, which makes sense when doing the initial load of the home screen (which is probably one of the bigger screens in Secret Server in terms of size).

SSH Key Verification Overview

SSH key verification is a process used to ensure that the SSH key you are connecting with belongs to the intended server or host. It is an essential security measure to prevent man-in-the-middle (MITM) attacks, where an attacker can intercept the connection and impersonate the server or host.

Here's how SSH key verification works:

1. **Host Key Fingerprint**: When you connect to an SSH server for the first time, the server sends its public host key to the client (your computer). This public key has a unique fingerprint or hash, which is displayed on the client's terminal.
2. **Verifying the Fingerprint**: You should compare the displayed fingerprint with the known and trusted fingerprint of the server. This trusted fingerprint can be obtained from the server administrator or from a trusted source, such as the server's website or documentation.
3. **Accepting or Rejecting the Key**: If the displayed fingerprint matches the trusted fingerprint, you can choose to accept and add the server's public key to your client's known-hosts file. This file stores the trusted public keys of the servers you have connected to before.
4. **Future Connections**: On subsequent connections to the same server, the client will check if the server's public key matches the one stored in the known_hosts file. If the keys match, the connection is allowed. If the keys do not match, the client will warn you about a potential MITM attack, and you can choose to accept the new key or abort the connection.

SSH key verification is crucial because it ensures that you are connecting to the intended server and not an impersonator. If an attacker intercepts the connection and tries to impersonate the server, the attacker's public key will not match the trusted key stored on your client, and the connection will be rejected or flagged as potentially insecure.

It is important to note that SSH key verification should be done carefully, especially when connecting to a server for the first time. You should verify the fingerprint through a trusted channel (such as, contacting the server administrator or checking the server's official documentation) to ensure that you are not accepting a rogue key from an attacker.

SSH key verification is an essential security practice that helps protect against MITM attacks and ensures the integrity and confidentiality of your SSH connections.

Server SSH Key Verification

Host SSH key verification is supported for use with heartbeat, proxied launchers, password changers, and discovery. Host SSH key verification can be used to ensure that the machine you are connecting to is a trusted host. Host SSH key verification will not pass credentials to the target machine unless the public key digest matches the SHA1 digest that Secret Server has on file. This helps prevent man-in-the middle attacks.

How to Map a Server SHA1 Digest to a Secret

To configure host SSH key verification:

1. go to Secret Templates and add a field for the host's SSH key digest.
2. Click **Configure Extended Mappings**.
3. Add a "Server SSH Key" mapping to your newly created SSH key digest field.
4. On your secrets, add the SSH Key digest of the hosts to your digest field. Verification takes effect the next time you connect to the host.

Heartbeat

If no "Server SSH Key" mapping exists for the secret or if the mapped digest field is blank, the digest will not be checked. If a digest is mapped and present and it does not match, then heartbeat will fail with a "UnableToValidateServerPublicKey" error. The heartbeat log will show the expected and actual values for the SHA1 digest.

Password Changing

If no "Server SSH Key" mapping exists for the secret or if the mapped digest field is blank, the digest will not be checked. If a digest is mapped and present and it does not match, then the password change will fail. The Remote Password Changing log will show the expected and actual values for the SHA1 digest.

Non-Proxied Launcher

When launching PuTTY, it displays a message if the server's public key digest is not yet trusted.

Proxied Launcher

If no "Server SSH Key" mapping exists for the secret or if the mapped digest field is blank, the digest will not be checked. If a digest is mapped and present and it does not match, then a message will be written to PuTTY displaying the expected and actual values for the SHA1 digest. The credentials from the secret will not be passed to the target machine.

SSH Script Dependencies

SSH Script dependencies now have a "Server Key Digest" field. When this field is blank, the server's digest will not be checked. When it is filled in, if it does not match, an error is returned indicating the expected and actual values from that server. No credentials are passed to the target machine unless the digest matches.

Unix Account Discovery

To validate SHA1 server digests for Unix account discovery, create a file named `KeyDigests.txt` in the root of the Secret Server website. Each line should contain an IP address or other computer identifier, a comma, and then the SHA1 digest (see example below). When the file exists and has data, all machines to be scanned must match one of the SHA1 hashes in the file. Any computers that do not match will still show up on the Discovery Network View page, but authenticated scanning will not take place (no credentials will be passed to the machine, and accounts will not be retrieved from the machine).

Sample `KeyDigests.txt`:

```
192.168.1.5,7E:24:0D:E7:4F:B1:ED:08:FA:08:D3:80:63:F6:A6:A9:14:62:A8:15
apollo,7A:25:AB:38:3C:DD:32:D1:EA:86:6E:1C:A8:C8:37:8C:A6:48:F9:7B
```

Be sure that your digest value which you input into `KeyDigests.txt` is not an MD5 value. MD5 values are 32 bytes, whereas a SHA1 is 40. So the above is correct whereas if you obtain a digest of your public key file and the result has 32 bytes, this is likely an MD5. One command which could be used to obtain the SHA1 digest of the public key file for use in `KeyDigests.txt` is:

```
awk '{$print $2}' id_xxx.pub | openssl base64 -d -A | openssl sha1
```

After which, you should add `:` to every two characters to the output such that it matches the above. We will try to connect to the keys which we have the strongest preference for first in the event of multiple keys and it is legal to have multiple digests for the same IP address or hostname in the file. In the event of multiple keys on your system, it is usually correct to get the digest from `/etc/ssh/ssh_host_rsa_key.pub` and put it into `KeyDigests.txt` if this public key exists.

SSL and Secret Server

Secret Server employs SSL (Secure Sockets Layer) to ensure that all communication between the user's web browser and the Secret Server application is encrypted, providing a secure channel for data transmission. By using SSL, Secret Server protects sensitive information such as passwords, secrets, and user credentials from being intercepted by unauthorized parties during transit. SSL also helps in verifying the identity of the server, mitigating the risk of man-in-the-middle attacks. Administrators can enforce SSL by enabling the "Force HTTPS/SSL" option in the Secret Server configuration, ensuring that all access to the application is conducted over HTTPS. Additionally, Secret Server supports HTTP Strict Transport Security (HSTS) to further enhance security by instructing browsers to only interact with the server over secure connections.

Installing Self-Signed SSL Certificates

Overview

An SSL (Secure Sockets Layer) certificate greatly enhances the security between the user's browser and the server your Secret Server is installed on. It encrypts all data between the server and the client's browser so if an attacker were to look at the data being transmitted between the two, they would not be able to decipher it.

SSL is required when using Integrated Windows Authentication.

Obtaining an SSL Certificate

You can get a certificate from various companies such as Thawte or VeriSign. If you already obtained a certificate from one of them, please follow their instructions for installing their certificates.

> Delinea does **not** provide certificates.

Installing a Self-Signed Certificate

You can create your own certificate for trial or sandbox environments:

> This requires IIS 7 or later.

Task One: Generate an IIS Self-Signed Certificate

1. Open IIS manager (**inetmgr**) on your Web server.
2. Click on the server node (one of the root nodes) in the left panel.
3. Double-click the **Server certificates** icon.
4. Click the **Create Self-Signed Certificate** link in the **Actions** panel. The Specify Friendly Name dialog box appears.
5. Type any name you desire in the **Specify a Friendly name for the certificate** text box.
6. Click the **OK** button. You now have an IIS self-signed certificate that is valid for one year. It appears under the Server Certificates panel. The certificate common name (Issued To column) is the host name of the machine running the site.

Task Two: Bind the Self-Signed Certificate to the IIS Site

1. In IIS Manager, click the server you want to bind to on the **Connections** panel tree.
2. Drill down to **Sites > Default Web Site**.
3. Click the **Bindings...** link in the **Actions** panel. The Site Bindings dialog box appears.
4. Click the **Add...** button. The Edit Site Binding dialog box appears.
5. Click the **Type** dropdown list and select **https**.
6. Click the **SSL certificate** dropdown list to select the certificate you just created.
7. Click the **OK** button. You return to the Site Bindings dialog box, where the HTTPS binding now appears.
8. Click the **Close** button. The dialog box closes.

Task Three: Test the Self-Signed Certificate

1. In a browser, go to the Website using the certificate. You should see a warning that there is an issue with the site's security certificate—specifically, the security certificate was issued for a different website's address. This occurs because IIS uses the server's name as the common name when using a self-signed certificate, which

usually does not match the hostname to access the site in your browser.

2. To access the website, click the "continue to the website" link or button. You will have to do this each time you access the site. Because this is a test environment, this should not be an issue.

Note: It is possible to remove the warning by adding the self-signed certificate to the trusted root certificate authorities, but that is beyond the scope of this instruction.

Trusting an SSL Certificate on a Client Machine

For public websites, only SSL certificates issued by trusted authorities are recognized as valid. Self-signed certificates used only within a company or domain might generate security warnings but these can be ignored. The same is true of self-signed certificates installed on a server for the Secret Server website. However, these security warnings can also interfere with the use of the Secret Server Launcher and Web Password Filler. To resolve these issues, install the certificate on the client machine, either through your browser or Certificates snap-in.

To enable trust in the Secret Server self signed certificates, following these steps:

Step 1: Compare Host Names

Make sure that the host to which the certificate is issued is the same as the host name for your Secret Server website:

1. Open your browser and navigate to Secret Server.
2. Click **Continue to this website** if you are prompted.
3. Click the **Certificate Error** icon next to the navigation bar.
4. Click the **View certificate** button. The value next to **Issued to** should match the host name for your website. For example, if your website is `https://www.mydomain.local/SecretServer`, it should say **Issued to:**`www.mydomain.local`. If these fields do not match, the client will not be able to fully trust the certificate.

Step 2: Transfer a copy from your server to the client computer

Obtain a copy of the certificate file and transfer it to the client computer:

1. On the server where Secret Server is installed, find **Run** from the start menu or screen and type in `mmc`, then click the **Enter** button.
2. From the **File** menu, select **Add/Remove Snap-in**.
3. Select the **Certificates** snap-in, then click the right arrow button to add it.
4. In the window that appears, select **Computer Account**.
5. Select **Local Computer**.
6. Click **Finish**. You should now see the **Certificates (Local Computer)** node.
7. Expand the **Personal** folder and then the **Certificates** folder under it.
8. Right-click the certificate that Secret Server uses.
9. Click **All tasks**.
10. Select **Export**.

11. Keep clicking the **Next** button to accept defaults in the wizard.
12. Type in a filename.
13. Click the **Finish** button. The certificate has now been exported.

> If you have Firefox, the certificate can be saved to your client computer by viewing and exporting it after navigating to the website.

Step 3: Install the certificate on the client computer

1. On the client computer, find **Run** from the start menu or screen and type in `mmc`, then press the **Enter** button.
2. From the **File** menu, select **Add/Remove Snap-in**.
3. Select the **Certificates** snap-in, then click the right arrow button to add it.
4. In the window that appears, select **My user account**.
5. Click the **Finish** button.
6. Expand the **Trusted Root Certification Authorities** folder.
7. Right-click the **Certificates** folder and select **All Tasks > Import**.
8. Click **Next** and **Yes** to accept default settings for all steps of the wizard.
9. When prompted for the certificate file, select the file you saved in the previous Step 2.

> You may need to re-open your browser and browse to Secret Server once more to see the change reflected on the client machine.

Multi-Factor Authentication

For added security, you can add a second layer of authentication, called multi-factor authentication (MFA) or two-factor authentication (2FA), when users access the system. This section discusses several options.

MFA on Secret Access

MFA on Secret Access is only available with Secret Server Cloud on the Delinea Platform. To upgrade from stand-alone Secret Server Cloud, please contact your Delinea sales or renewal representative. You can read about the upgrade process here.

Applications for Soft Token Two-Factor Authentication

The name of the Secret Server soft token two-factor setting for Time-based One Time Password (TOTP) is “TOTP Authenticator." Any TOTP application that uses the TOTP RFC6238 algorithm, such as such as Microsoft Authenticator, will work with the Secret Server TOTP Authenticator. Details on the TOTP RFC6238 standard can be found at TOTP: Time-Based One-Time Password Algorithm.

> Google Authenticator support started in Secret Server version 8.6.

Duo Security Authentication

> Using this method of two-factor authentication requires that you have an active account for Duo Security.

> Secret Server supports using Duo Security as a second factor of authentication. See below for setup instructions.

> For more information on Duo and Secret Server, see the DelineaSecret Server and Duo page.

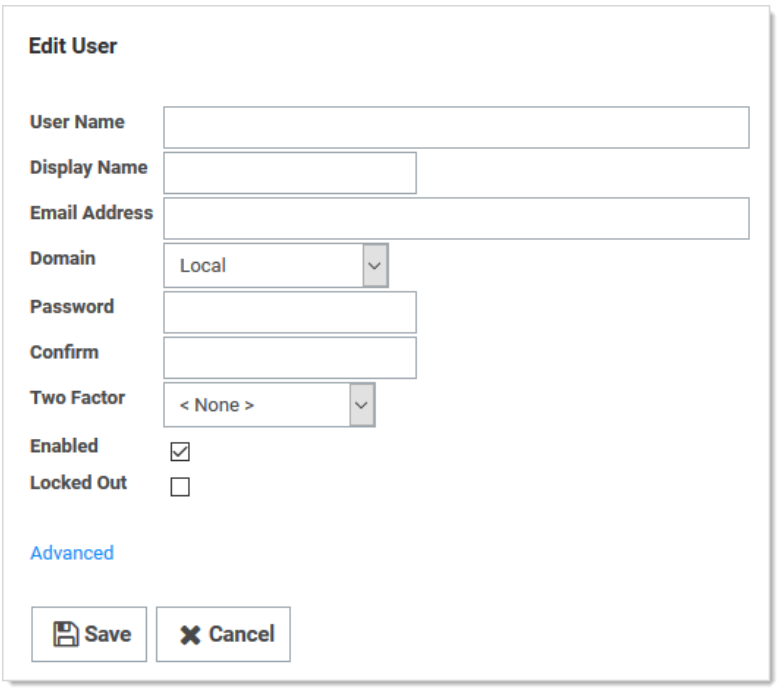
Task 1: Create a Duo Application Representing Your Secret Server (Admin)

1. Sign up for a new Duo account, or log in to an existing one at Duo Security.
2. Under **Applications**, create a new application of the **DelineaSecret Server** type. Name the application as you wish.
3. Record the API hostname, integration key, and secret key from the new Duo application you just created.

Task 2: Configure Secret Server to Use Duo (Admin)

> Because Duo is a service, the Secret Server instance must have outbound access (TCP port 443) to reach the API host to work. If there is a firewall rule preventing access to Duo's servers, two factor authentication will not work.

1. Open Secret Server.
2. From the **Admin** menu, select **Configuration**.
3. Click the **Login** tab, and then click **Edit**.
4. Select the **Enable Duo Integration** check box.
5. Enter the **API Hostname**, **Integration Key**, and **Secret Key** values.
6. Click the **Save** button.
7. Go to **Admin > Users** to create a test user. The Users page appears.
8. Click the **Create New** button. The **Edit User** page appears:



9. Click the **Two Factor** dropdown list and select **Duo**.
10. Type or select the other parameters for the new user.
11. Log on as the test user. If there are multiple two-factor devices available, you will be prompted to select one. If you are un-enrolled you will be given a link to perform self-enrollment. You are contacted via the Duo app, SMS, or a phone call for the second factor.
12. Add or configure actual users one at a time or by using bulk operations.

Task 3: Setting up Duo (User)

1. Log on to Secret Server.
2. After successful authentication, a new screen appears with the option to select a method to authenticate with.
3. Select one of the options (**Duo Push**, **Send SMS**, or **Phone)**, depending on your setup with Duo) and complete the selected authentication process to log in.

Email Two-Factor Authentication

Secret Server requires that a connection to a SMTP server be properly configured to send out confirmation code emails. Enter the SMTP server information and an email address that is used to send notifications:

1. Click **Admin > Configuration**.
2. Click the **Email** tab.
3. Verify SMTP server availability with telnet using the command `telnet <your server name> 25`.

If virus protection is running, you may need to add a firewall rule to allow aspnet_wp.exe to send e-mails.

FIDO2 (YubiKey) Two-Factor Authentication Configuration

Overview

FIDO2

FIDO2 (Fast Identity Online, second edition) is an open authentication standard that uses physical devices for authentication. Delinea uses this standard for two factor authentication (2FA) with FIDO2 providing the second authentication after a normal password entry. Any FIDO2-enabled user attempting access to a Secret Server account **must** have a FIDO2 device in hand. The device eliminates many password-related issues, such as phishing and man-in-the-middle attacks. It also speeds up the login process over callback or texting 2FA.

YubiKey

YubiKey is a FIDO2-compliant product series from Yubico, a commercial company. We recommend two of their devices, the YubiKey 5 Series and the Security Key.

Configuration

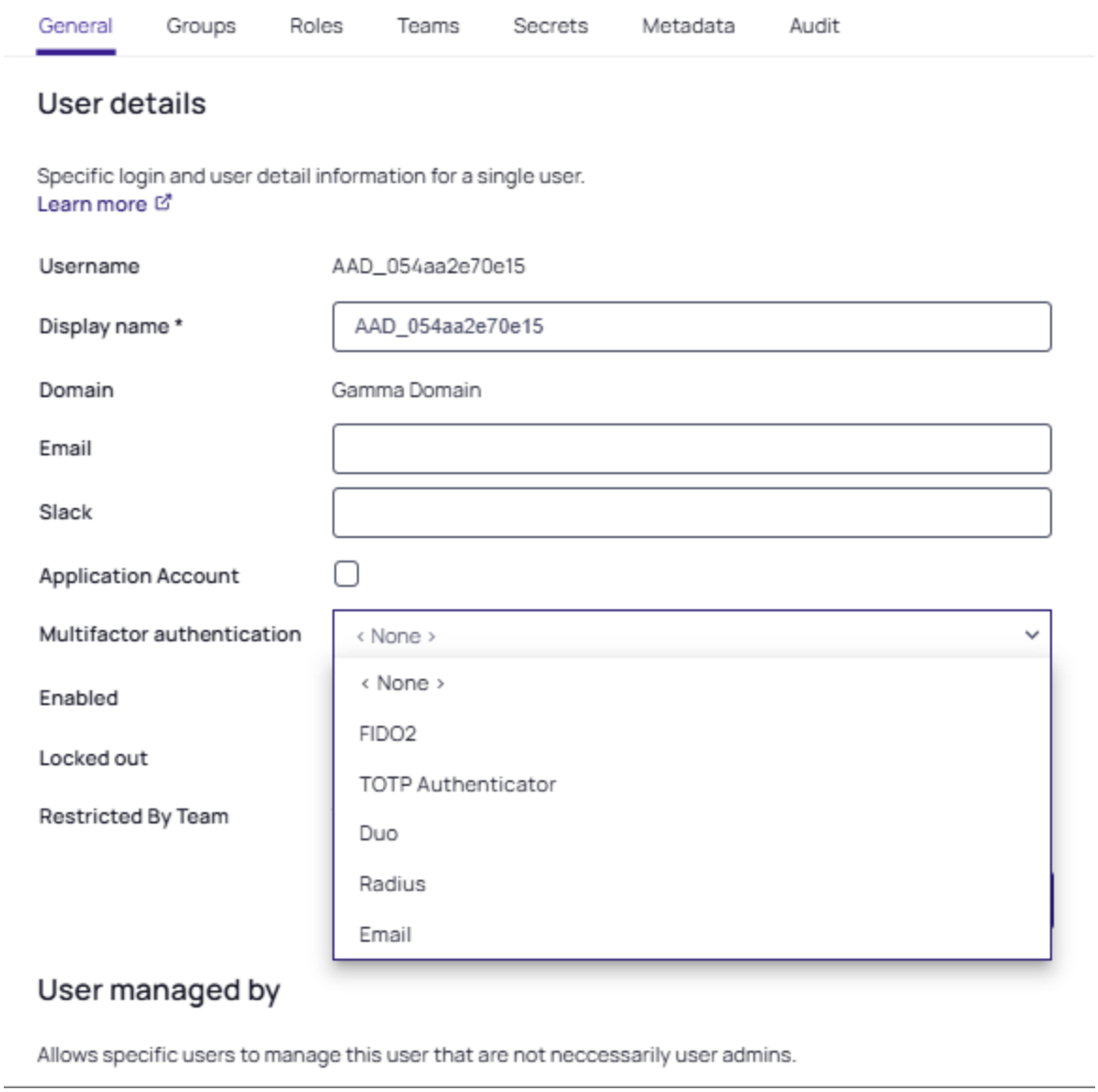
Prerequisites

- One FIDO2 device. We recommend the YubiKey series.
- A Secret Server Vault license or greater.
- Administer Users or User Owner permissions in Secret Server.
- A Firefox or Chromium browser, such as Google Chrome or Microsoft Edge.

Enabling FIDO2 for a Single User

1. In Secret Server, go to **Access > Users**. The **User Management** page appears.
2. Click on the user you wish to edit.
3. Click **Edit** in the **User Details** section at the top of the page (in the **General** tab).

4. Click on the **Multifactor Authentication** list and select **FIDO2**:

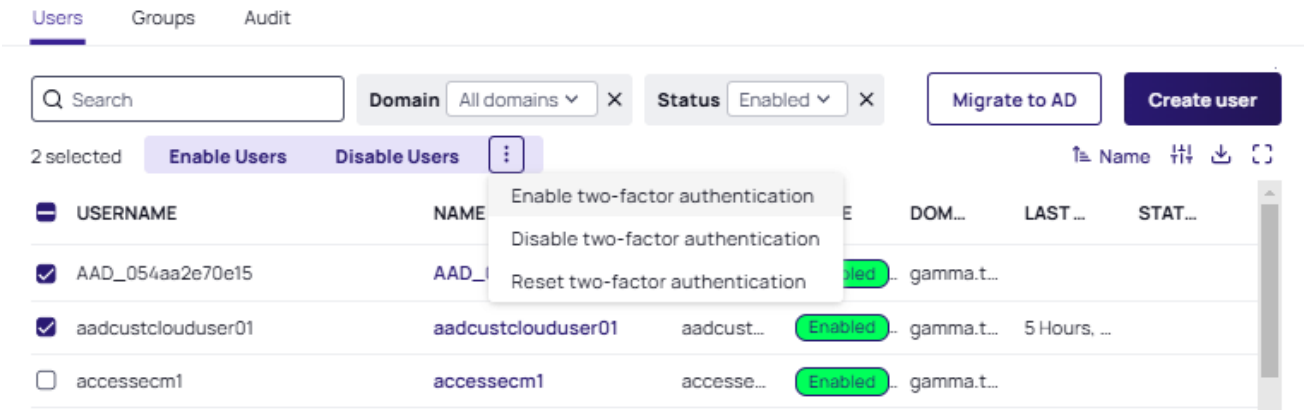


5. Click the **Save** button.

Enabling FIDO2 for Multiple Users

1. In Secret Server, go to **Access > Users**. The **User Management** page appears.
2. Select the check box next to each user you wish to include. The **Enable Users** link appears at the top alongside the **Disable Users**.

3. Click the three dots next to **Enable Users** and select **Enable Two-Factor Authentication** from the dropdown list:



4. In the **Enable Two-Factor Authentication** popup, select **FIDO2** in the **Two-Factor Authentication Provider** drop-down list.
5. Click the **Save** button. The **Bulk Progress** popup appears.
6. When the **Task Complete** message appears, click the **Close** button.

Disabling FIDO2 for Users

The process to disable FIDO2 for both single and multiple users is almost identical to enabling them. There are two differences:

- For a single user, select **<None>** for the **Multifactor Authentication** list on the **Edit User** page, and then**Save**.
- For multiple users:
  1. Select **Disable Two-Factor Authentication** in the drop-down list made available after selecting multiple user checkboxes in the **User Management** page, the **Disable two-factor authentication** popup appears.
  2. From the popup select FIDO2 before clicking **Save**. See the previous section for a visual example.

> Disabling FIDO2 2FA does **not** remove device registration information from Secret Server. If FIDO2 is re-enabled, the user can use the FIDO2 device without re-registering it.

Unregistering Users from FIDO2

Resetting FIDO2 serves to unregister existing users. There is no way to reverse this action, users will have re-register the FIDO2 device, even if it is the same device they used previously.

Resetting FIDO2 for both single and multiple users is very similar to enabling FIDO2 for multiple users. The only difference is that you select **Reset Two-Factor Authentication** in the drop-down list made available after selecting a single or multiple user checkboxes in the **User Management** page. See the image above for a visual example. The operation is exactly the same for single and multiple users.

Registering FIDO2 Devices (End-User Operation)

1. After an admin registers your user in Secret Server, you are prompted upon your next login to use either the "security key with the localhost" (Chrome) or that "localhost wants to register an account with one of your security keys. You can connect and authorize one now or cancel" (Firefox).

> Legacy Microsoft Edge is not supported. Edge Chromium, version 79 or higher, is required for FIDO2 support.

2. Insert your FIDO2 device into a USB port on the computer.
3. Activate it by touching the sensor on the device.
4. After a successful registration, you are **again** prompted with the same screen, which is authenticating the current session with the credentials that were just registered.
5. From then on, you will be prompted for your security key after a successful login. Once the key is authenticated, the Secret Server Dashboard appears.

> Only one FIDO2 device per user can be registered at any given time. The 2FA settings, however, can be temporarily disabled or reset in the case of a lost or forgotten FIDO2 device.

Auditing and Security

- Upon registration your FIDO2 Credential, FIDO2 Public Key JSON string, and the FIDO2 Counter are all stored in your user's audit log.
- Upon each successful FIDO2 authentication, the FIDO2 counter value is updated and noted in your audit log.

Troubleshooting and Issues

- If you encounter an error or do not complete authentication before the process times out, then you will be redirected back to the username and password login screen where the process can be reattempted.
- Authentication activities are logged in your audit log, to help with the tracking of any potential issue reproduction steps.
- System errors are logged in the Secret Server.log file in the Secret Server log directory.

RADIUS User Authentication

Secret Server allows the use of *Remote Authentication Dial-In User Service* (RADIUS) two-factor authentication on top of the normal authentication process for additional security needs. Secret Server acts as a RADIUS client that can communicate with any server implementing the RADIUS protocol.

Configuring RADIUS

Set up RADIUS on the **Login** tab of the **Configuration** page. This requires enabling RADIUS Integration, specifying the server address, the ports, and the RADIUS shared secret. The shared secret is a specific term for RADIUS clients and is not a reference to secrets in Secret Server.

You can customize the RADIUS "Login Explanation" to give users detailed instructions for entering their RADIUS information.

Once enabled, the **Test RADIUS Login** button appears on the **Login** tab for testing the communication with the RADIUS Server. If you have a failover RADIUS Server, you can specify it by clicking the **Enable RADIUS Failover** checkbox and entering the required information. If the primary RADIUS server cannot be accessed, the failover server is be used.

Enabling RADIUS for a User

After enabling RADIUS on your Secret Server, you must enable RADIUS two-factor authentication for each user on a per-user basis. On the **User Edit** page, type the **RADIUS User Name** for this user to match the RADIUS server. RADIUS can be enabled for new users by domain.

Enabling RADIUS Two-Factor Authentication

RADIUS authentication is handled by the web nodes, rather than the distributed engines, in both Secret Server Cloud and On-Premises environments.

Procedure

Secret Server allows the use of RADIUS two-factor authentication on top of the normal authentication process for additional security.

See the full RADIUS Integration Guide for additional information.

To configure RADIUS for the Secret Server instance:

1. Log on Secret Server with an account with "Administer Configuration" and "Administer RADIUS" permissions.
2. Navigate to **Administration menu**.
3. Type **RADIUS** in the search box and press **<Enter>**. The RADIUS Configuration page appears.
4. Click the **Edit** button.
5. Type the following as needed:
   - **RADIUS Login Explanation**: (custom message or instruction). Defaults to "Please enter your RADIUS passcode."
   - **RADIUS Default Username**: select the related RADIUS username from the dropdown. The default RADIUS username determines the credential Secret Server uses when attempting to authenticate against your RADIUS server. If your RADIUS server requires a UPN (User Principal Name), this setting should be

modified accordingly. The UPN is stored in the database. If a Username is used, Secret Server will send the SAM Account Name (domain\username) to the RADIUS server. This setting can be customized to meet the specific requirements of your RADIUS server by navigating to **Admin > Users** when configuring RADIUS as the two-factor authentication method.

- **RADIUS Client Port Range**: (default 1812) source ports, instructing Secret Server to send requests exclusively through the defined client port range. By entering a value of 0, you can configure Secret Server to use the ephemeral port range for outgoing requests.

  If your RADIUS server runs on the same machine as Secret Server, the client and server ports must be different.

- **RADIUS Server Port**: (default 1812 for RSA and 1812 for AuthAnvil).
- **RADIUS Server IP**: (IP address to your RADIUS Server). See RADIUS IP Addresses.
- Leave **Use Same RADIUS Shared Secret for All Users**: selected.
- **RADIUS Shared Secret**, which must match chosen RADIUS shared secret on your RADIUS Server. (Shared Secret is a RADIUS term and not related to any Secret Server secret.)

Attempt Silent Authentication: Silent answer is a new configuration option for RADIUS that allows setting the RADIUS response to a defined string value. This is to support push notification and other interactive variations in advanced RADIUS authentication configuration. The new setting replaces "Attempt User Password" and allows for sending the user password or another predefined string.

- **RADIUS Protocol**: select UDP or EAP-TTLS-PAP from the dropdown.
- **Time out (seconds)**: set the number of seconds for time out.
- **Enable Failover RADIUS Server**: enabling a failover RADIUS server will allow another server to fail over to.
- **Failover RADIUS Server Port**: enter the related failover server port.
- **Failover RADIUS Server IP**: enter the IP address of your failover RADIUS server.
- **Failover RADIUS Shared Secret**: enter the related secret for RADIUS failover server.
- **Failover Time Out (Seconds)**: set the number of seconds for RADIUS failover server time out.
- **Attempt Silent Authentication**: select User Password or Static Value. For Static Value, enter the value below to send to RADIUS as the password.
- **Enable RADIUS NAS-Identifier**: Check to enable and configure the NAS-Identifier that will be sent with the RADIUS Access-Request. This Attribute contains a string identifying the NAS originating the Access-Request. It is only used in Access-Request packets. Either NAS-IP-Address or NAS-Identifier MUST be present in an Access-Request packet.

> Note that NAS-Identifier must not be used to select the shared secret used to authenticate the request. The source IP address of the Access-Request packet must be used to select the shared secret.

- **Disable RADIUS NAS-IP-Address attribute**: check to disable. This Attribute indicates the identifying IP Address of the NAS which is requesting authentication of the user, and should be unique to the NAS within the scope of the RADIUS server. NAS-IP-Address is only used in Access-Request packets. Either NAS-IP-Address or NAS-Identifier must be present in an Access-Request packet.

  > Note that NAS-IP-Address MUST NOT be used to select the shared secret used to authenticate the request. The source IP address of the Access-Request packet must be used to select the shared secret.

6. Click the **Save** button.

To test RADIUS settings:

1. Click the **Test RADIUS Login** button. A popup appears.
2. Type the RADIUS username and password.
3. Click the **OK** button.
4. After enabling RADIUS on Secret Server, you must enable RADIUS two-factor authentication for each user:
   a. Sign into an account with "Administer Configuration" and "Administer RADIUS" permissions.
   b. Navigate to **Administration** > **Users**. The Users page appears.
   c. Select the desire user.
   d. Click the **Edit** button.
   e. Click to select the **RADIUS Two Factor Authentication** check box.
   f. Type the username in the **RADIUS Username** text box.

      NOTE: Secret Server defaults this value to its username. If you wish to use this default name, it must match the username on the RADIUS server.
   g. Review the settings and click **Save**.
   h. Repeat these steps for each user that needs to use RADIUS.

RADIUS IP Addresses

Please see the Secret Server Cloud Architecture Documentation for a listing of IP addresses.

TOTP

Secret Server supports using any type of soft token or mobile application authentication using the *Time-Based One-Time Password* (TOTP) RFC6238 algorithm. TOTPs are typically generated and authenticated by a mobile application using an algorithm that incorporates the current time to ensure that each one-time password (OTP) is unique. TOTP applications include Authy, Google Authenticator, and Microsoft Authenticator.

Secret Server can also serve as an OTP generator, providing TOTP authentication for RPC and launchers. The soft token two-factor function in Secret Server is the "TOTP Authenticator" and you can use any application that uses the TOTP RFC6238 standard (details on the standard can be found at the IETF Tools website. An example of a TOTP application that works with Secret Server soft token two-factor authentication is Microsoft Authenticator.

> The same 32-character key can generate different TOTP outputs based on the hashing algorithm used (SHA1, SHA256, or SHA512). It is essential to configure the Secret Server template to use the same hashing algorithm as the external system providing the key. This ensures that the generated TOTP codes are accurate and can be successfully authenticated.

Enabling TOTPs for Launchers

Most commonly, time-sensitive one-time passwords (TOTPs) are generated by a mobile application, such as Google Authenticator or Microsoft Authenticator. Additionally, Secret Server can be used as the TOTP generator for RPC or launchers. Both the secret and the secret template require configuration for this use.

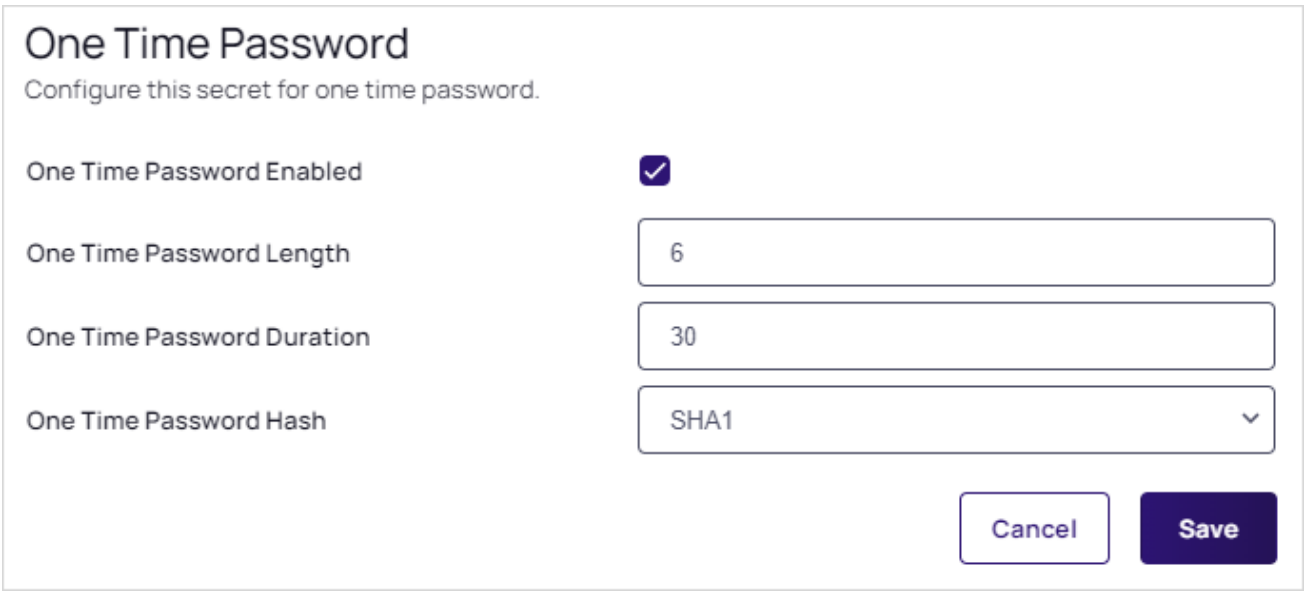
Any template type can have a TOTP code generated in the secret once TOTP is enabled in the secret template and in the secret itself. Only web password secrets have autofilling TOTPs. That is because that is based on the only template type Web Password Filler (WPF) is looking at. See Autofilling OTP in Using WPF for details.

> When configuring TOTP on a Secret Server template, make sure that the hashing algorithm specified in the template matches the algorithm used by the external system providing the key. The same 32-character key can produce different TOTP outputs depending on whether SHA1, SHA256, or SHA512 is used. Failure to match the hashing algorithms can result in incorrect TOTP codes and authentication failures.

Secret Template Setup

To enable TOTP on a Secret Server template:

1. Go to **Administration button > Secret Templates**.
2. Select the desired template, and click the **Edit** button. The Secret Template Designer appears.
3. Navigate to the **One Time Password** section of the page, and click the **Edit** button.
4. Click to select the **One Time Password Enabled** check box. This enables the option with default settings:



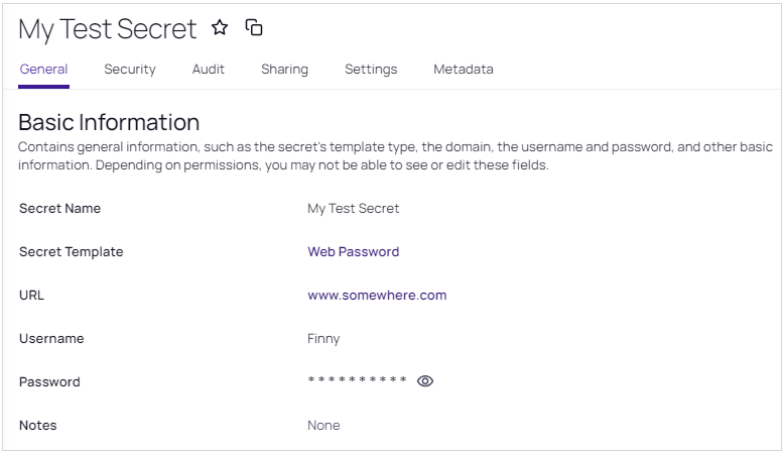
These are the values that most one-time password instances, such as Google and Microsoft Authenticator, use today. If you use these settings with another OTP provider and are unable to successfully use generated codes to authenticate, please review their documentation and adjust these settings as required.

5. Save the secret template. Any web password secret based upon this template can now use TOTP.

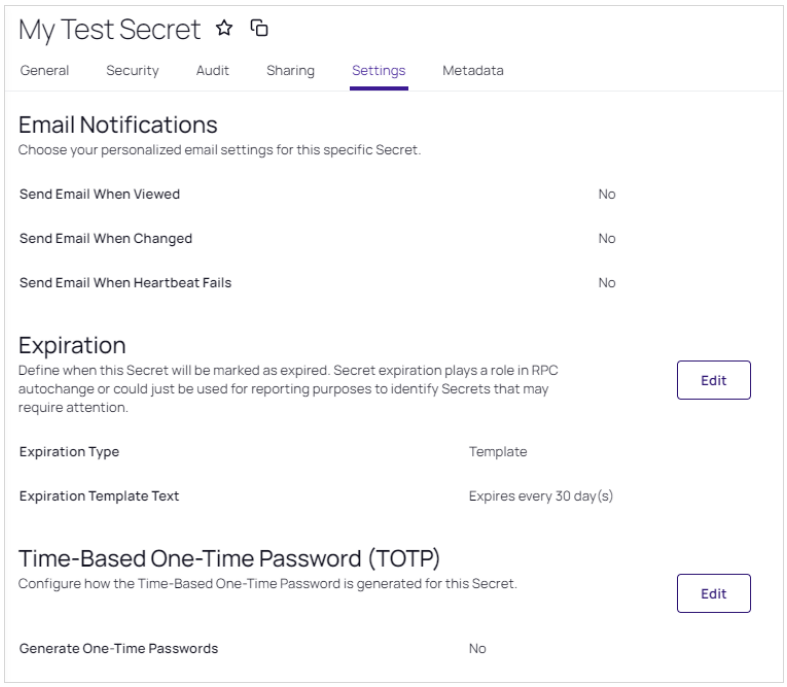
TOTP Secret Setup

Once a secret template is set up for TOTP, each secret based on that template also needs to be set up:

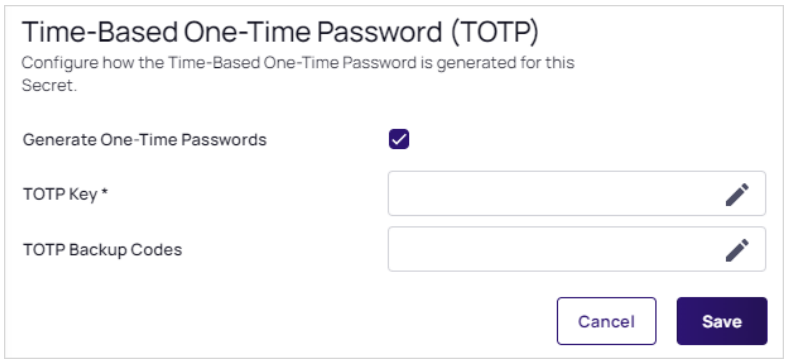
1. Click the **Secrets** menu item in the dashboard.
2. Open the desired secret.



3. Click the **Settings** tab:



4. Click the **TOTP** section's **Edit** button
5. Click to select the **Generate One-Time Passwords** check box in the **TOTP** section. This exposes two text boxes:



6. Type the TOTP key in the **TOTP Key** text box. The TOTP Key is generated by the OTP-protected asset when you set up your account to use TOTP. Usually, you are prompted with a QR barcode that you can scan with a mobile device, or you can expose the key that the QR code represents by selecting the **Manual Setup** link when performing the initial TOTP setup for a user. The TOTP Key is found in the **Key** field. This text string is the value that is placed into the TOTP Key field.

> Treat the TOTP key and backup codes like you would any other password! If anyone obtains the key, it can be used to set up a valid TOTP generator for that account on any device, allowing that person to bypass the protection. Similarly, the backup codes allow users to temporarily bypass protection.

> If you have an account that has been TOTP protected and you did not save the TOTP key upon creation, you must deactivate TOTP on that account and then reactivate it to retrieve the TOTP key to set up Secret Server.

7. Type the TOTP backup codes in the **TOTP Backup Codes** text box. The TOTP Backup Codes are often presented to a user while initially setting up an account for TOTP. These backup codes are single-use codes for use if a TOTP generator is not available or working. Again, these codes will be valid and allow the holder to get past the two-factor authorization to access an account, so protect them as you would a password!

Enabling TOTP for Secret Server Users

1. From the **Admin** menu, select **Users**.
2. Select the check box beside each user to enable two-factor authentication for.
3. From the **< Select Bulk Operation >** drop-down menu, select Enable TOTP Auth Two Factor.
4. Click **OK** in the dialog that appears, confirming the operation.
5. The user(s) are now required to complete the soft token setup with a mobile device the next time they log into Secret Server. See Onboarding the Mobile App for details on the account and mobile app setup that follow.

Disabling TOTP for Users

To disable soft token two-factor authentication, follow almost the same process as enabling soft token two-factor authentication for a user, select **Disable TOTP Auth Two Factor** from the bulk operation drop-down menu instead of **Enable TOTP Auth Two Factor**.

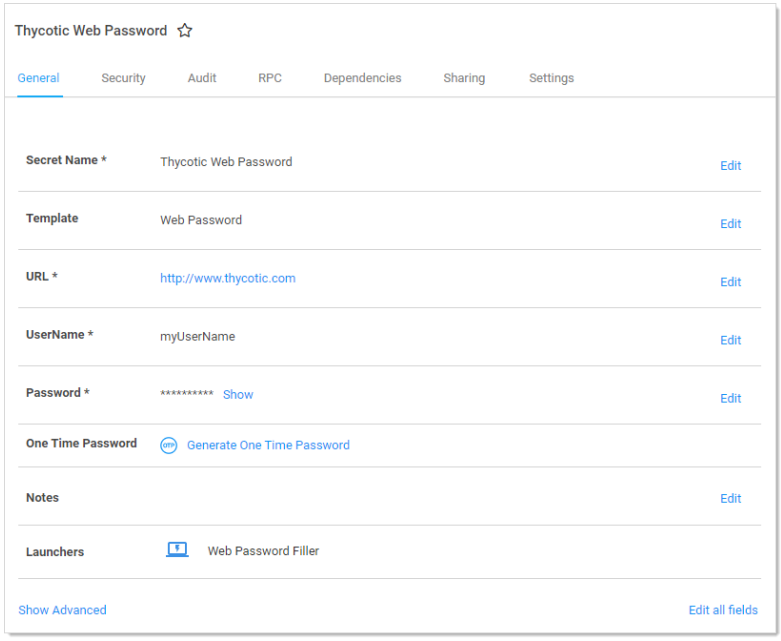
Resetting TOTP for Secret Server Users

1. From the **Admin** menu, select **Users**.
2. Select the check box beside the user to reset two-factor authentication for.
3. Click select **Reset TOTP Auth Two FactorFrom** on the **< Select Bulk Operation >** drop-down menu.
4. Click **OK** in the dialog that appears, confirming the operation.
5. The user is now required to complete the soft token setup with a mobile device the next time they log into Secret Server. See **User Setup of Soft Token Two-Factor Authentication** for further details on the account and mobile app setup that follow.

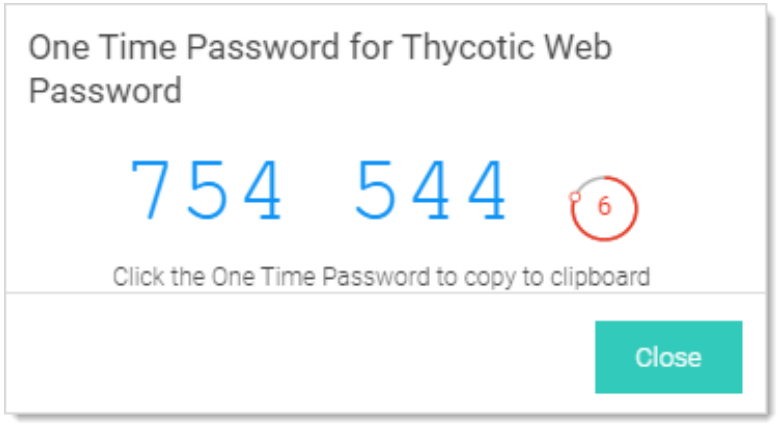
Viewing a TOTP for a Web Secret

To view or copy the TOTP generated for an account:

1. Navigate to and open the desired secret.
2. Click the **General** tab:



3. Click the **Generate One Time Password** link next to the **One Time Password** setting.
4. A dialog box appears with an OTP:



5. Click the OTP to copy it to the clipboard.
6. Click the **Close** button.

> The "Generate One Time Password" link also appears on the preview pane when you click a secret on the All Secrets page.

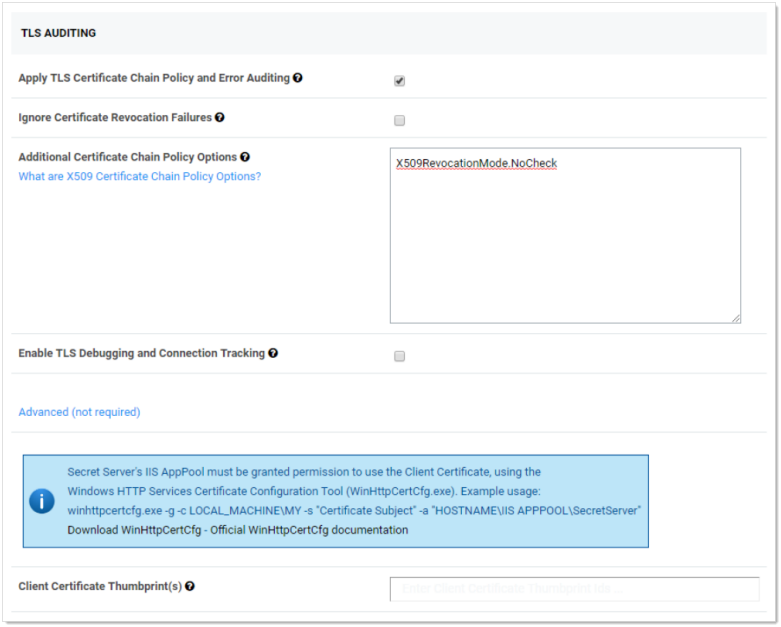
X.509 Certificate Security Chain Options

Starting with version 10.4, Secret Server allows you to define a policy for validating X509 Certificates. This applies to all Active Directory domains using LDAPS. It also applies to any connections to syslog servers over TLS. Certificates that do not meet the policies specified in Secret Server are rejected, denying connections to the server. All certificate validation failures are logged in the security audit log, which is available by going to **Admin > See All** and then **Security Audit Log**.

Setting the Certificate Verification Policy

To set a verification policy:

1. Go to **Admin > Configuration**.
2. Click the **Security** tab.
3. Click the **Edit** button
4. Click to select the **Apply TLS Certificate Chain Policy and Error Auditing** check box. The TLS Auditing options appear:



5. To change the policy, type a semi-colon delimited list of policy options in the **Additional Certificate Chain Policy Options** text box. To use a policy, enter the `<full_enumeration_name>.<enumeration_item>`. For example, to validate the entire certificate chain, add `X509RevocationFlag.EntireChain` to the semi-colon delimited list of options. See Certificate Validation Options for details.
6. If you wish to ignore certificate revocation warnings and allow revoked certificates, click to select the **Ignore Certificate Revocation Failures** check box.

Certificate Validation Options

The following Microsoft enumerations are the available certificate chain policy options. For detailed descriptions of each option, see the linked documentation.

X509RevocationMode

Specifies the mode used to check for X.509 certificate revocation.

Enumeration Item	Value	Description
NoCheck	0	No revocation check is performed on the certificate.
Offline	2	A revocation check is made using a cached certificate revocation list (CRL).
Online	1	A revocation check is made using an online certificate revocation list (CRL).

See X509RevocationMode Enum for details.

X509RevocationFlag

Specifies which X.509 certificates in the chain should be checked for revocation.

Enumeration Item	Value	Description
EndCertificateOnly	0	Only the end certificate is checked for revocation.
EntireChain	1	The entire chain of certificates is checked for revocation.
ExcludeRoot	2	The entire chain, except the root certificate, is checked for revocation.

See X509RevocationFlag Enum for details.

X509VerificationFlags

Specifies conditions under which verification of certificates in the X.509 chain should be conducted. These values can be bitwise combined to indicate multiple flags.

Enumeration Item	Value	Description
AllFlags	4095	All flags pertaining to verification are included.
AllowUnknownCertificateAuthority	16	Ignore that the chain cannot be verified due to an unknown certificate authority (CA).

Enumeration Item	Value	Description
IgnoreCertificateAuthorityRevocationUnknown	1024	Ignore that the certificate authority revocation is unknown when determining certificate verification.
IgnoreCtlNotTimeValid	2	Ignore that the certificate trust list (CTL) is not valid, for reasons such as the CTL has expired, when determining certificate verification.
IgnoreCtlSignerRevocationUnknown	512	Ignore that the certificate trust list (CTL) signer revocation is unknown when determining certificate verification.
IgnoreEndRevocationUnknown	256	Ignore that the end certificate (the user certificate) revocation is unknown when determining certificate verification.
IgnoreInvalidBasicConstraints	8	Ignore that the basic constraints are not valid when determining certificate verification.
IgnoreInvalidName	64	Ignore that the certificate has an invalid name when determining certificate verification.
IgnoreInvalidPolicy	128	Ignore that the certificate has invalid policy when determining certificate verification.
IgnoreNotTimeNested	4	Ignore that the CA (certificate authority) certificate and the issued certificate have validity periods that are not nested when verifying the certificate. For example, the CA cert can be valid from January 1 to December 1 and the issued certificate from January 2 to December 2, which would mean the validity periods are not nested.
IgnoreNotTimeValid	1	Ignore certificates in the chain that are not valid either because they have expired or they are not yet in effect when determining certificate validity.
IgnoreRootRevocationUnknown	2048	Ignore that the root revocation is unknown when determining certificate verification.
IgnoreWrongUsage	32	Ignore that the certificate was not issued for the current use when determining certificate verification.
NoFlag	0	No flags pertaining to verification are included.

See X509VerificationFlags Enum for details.

Troubleshooting

If you enable certificate policy validation and logging, you may have server connections rejected due to certificates that violate the set policies. These errors are recorded in the security audit log. If the information logged there is not enough to determine why a certificate was rejected, you can get additional log details by enabling TLS Debugging. This adds detailed information to the logs about each certificate checked.

Due to the possibility of exposing sensitive information in the logs, TLS debugging requires two steps to enable:

1. Click to select the **Enable TLS Debugging and Connection Tracking** check box.
2. Change the global logging level to DEBUG. To do this, edit the `web-log4net.config` file in the root folder of your Web application. Follow the comments in the file to comment out the current log level line (the default is INFO), and uncomment the line that sets the value to DEBUG.

> Only enable TLS debugging when you are actively troubleshooting a certificate validation issue. Disable this option when you are not to prevent logging of certificate details.

Backup and Disaster Recovery

Disaster recovery and backup in Secret Server are essential components designed to ensure the resilience and availability of critical data and systems. Secret Server supports both manual and scheduled backups of the database and IIS directory, providing a robust mechanism to safeguard against data loss. The platform allows for SQL mirroring and automatic failover, enhancing its disaster recovery capabilities.

Administrators can also export secrets to a CSV file as an additional precaution. In the event of a disaster, Secret Server's disaster recovery features enable quick restoration of the application and database, minimizing downtime and ensuring that privileged access remains secure and accessible. These measures collectively help organizations maintain operational continuity and protect sensitive information from various threats.

Secret Server Backup

> This topic only applies to **Secret Server On-Premises**.

Secret Server offers robust backup features to ensure data integrity and availability in case of system failures or disasters. It supports both manual and scheduled backups of the database and IIS directory, allowing administrators to configure automatic backups at specified intervals. These backups can be stored locally or on a network share, providing flexibility in storage options. Secret Server also supports SQL mirroring and automatic failover for enhanced disaster recovery. Additionally, administrators can export secrets to a CSV spreadsheet as an extra precautionary measure. The backup settings include options to specify file paths, set retention periods for operational logs, and configure notifications for backup failures, ensuring comprehensive backup management.

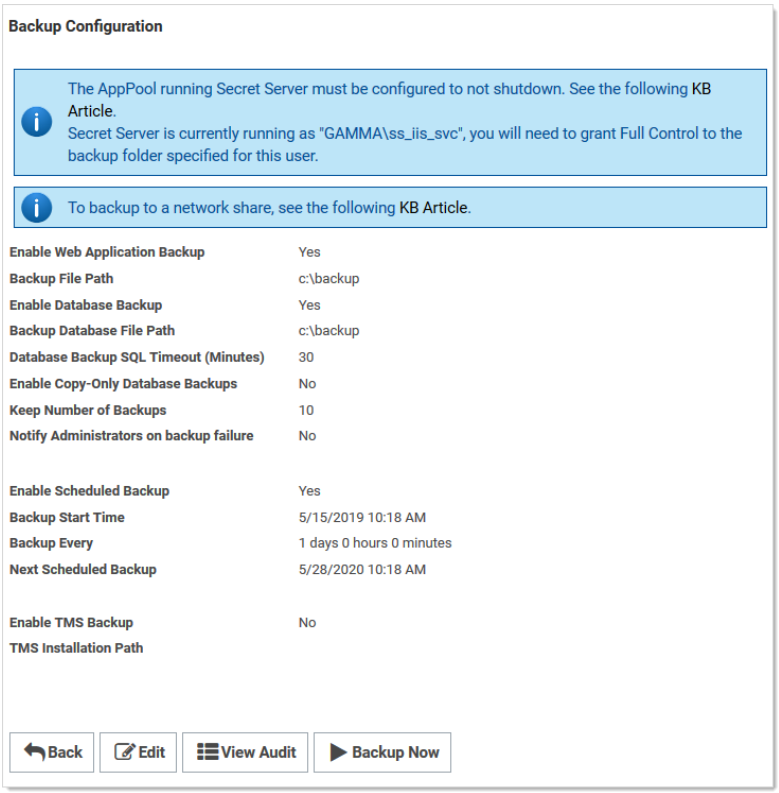
Backing up Secret Server to a Network Share

This topic only applies to **Secret Server On-Premises**.

Secret Server can be configured to backup to a network share instead of a local folder on the server. For example, you may want to do this such as when the Secret Server database (SQL) is located on a different server than the web application server (IIS).

To back up:

1. Ensure the Secret Server IIS Application Pool is running as a service account if it is not already. See Secret Server"Running the IIS Application Pool As a Service Account" on page 60.
2. Grant access to the network share (using Windows ACLs) to the account running the Secret Server IIS Application Pool (so that Secret Server can backup the application folder and zip it to the network share).
3. Grant access to the network share (using Windows ACLs) to the account running Microsoft SQL Server service. (so that Microsoft SQL Server can backup the Secret Server database to the network share). You can change the service account running Microsoft SQL Server by going to SQL Server Configuration Manager.
4. Go to **Admin > Backup**. This may require you to go to **Admin > All** and search for **Backup**.



5. Note that the two file paths are from two different perspectives–Backup File Path is from the ASP.N ET application server and Backup Database Path is from the Microsoft SQL Server (these may be on the same box in your environment, or they might not be depending on how you have configured Secret Server).

6. Click the **Edit** button.
7. Type the Secret Server backup path, such as \\server01\backup\secretserver\, in the **Backup File Path** text box.
8. Type the database backup path in the **Backup Database Path** text box.
9. Click the **Save** button.

Backup Settings

This topic only applies to **Secret Server On-Premises**.

Overview

The following configuration options are available on the **Tools > Backup** page of Secret Server:

- **Backup Database File Path:** This folder must be accessible by the SQL server and stores the `database.bak` file. See File Path Settings.
- **Backup File Path:** This directory must exist on the Web server and stores the zip file of the application directory. See File Path Settings.
- **Database Backup SQL Timeout (Minutes):** Number of minutes that Secret Server waits for the database backup to complete successfully before timing out.
- **Enable Scheduled Backup:** Enables automatic backups on a set schedule.
- **Days to Keep Operational Logs**: Sets the period to keep backup-related logs that might contain PII. Secret Server automatically deletes logs older than that (in days).
- **Keep Number of Backups:** Number of previous backups to keep.
- **Notify Administrators on Backup Failure:** Users with the Administer Backup role permission are notified if the backup fails.

File Path Settings

There are two file path settings on the **Admin > Backup** page (`ConfigurationBackup.aspx`). The "Backup File Path" setting corresponds to the application backup. The "Backup Database Path" setting corresponds to the SQL server backup.

Generally, the "Backup File Path" setting can be set to a path local to the application server for backing up of application files. If Secret Server is running under an account that does not have permission to write to a local path, then a network share can be used. If the SQL server is located on the same server as the Web application server, the "Backup Database File Path" setting can be set to a local path.

If the SQL server is not located on the same server as the Web application server then a network share should be used. The account under which SQL server service is running either must have modify rights to that path or must be a member of a group with modify rights to that path. You must use UNC (Universal Naming Convention) notation to write to a network path. For example: `\\TESTVM0\c$\backupDirectory`.

If you get an error stating "Cannot open backup device... Operating system error 3," this is often due to an invalid path value.

For Secret Server to delete old database backups, the backup database path must also be accessible by the Secret Server Application Pool account.

Backup Folder Permissions

This topic only applies to **Secret Server On-Premises**.

From the Backup Administration page, specify the correct directory paths for the IIS Secret Server file directory and the database backups to be stored. The backup path must be local to the server where the Secret Server database or file directory exists. The directories must also have the proper permissions to allow Secret Server to automatically store backups at those locations. The account that requires permission is displayed as an alert on the Backup page.

Common Backup Errors

This topic only applies to **Secret Server On-Premises**.

Cannot open backup device... Operating system error 3

This is often due to an invalid path value for the "Backup Database File Path" setting. For more information on the proper values for this setting, see "File Path Settings" on the previous page.

Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding.

This is often due to an overly-large database. The Secret Server database likely contains too many log entries. To clear these, within Secret Server, select System Log from the Administration menu. Click the "Clear" button below the data grid that contains the log entries. If the timeout occurs with the clear as well, an upgrade to the latest version should resolve this. If the timeout issue persists with the backup, additional SQL database clean-up may be necessary. Contact Delinea Support for instructions on shrinking the reserve database size.

The process cannot access the file... because it is being used by another process

The cause of this message is typically multiple backup threads running simultaneously with all attempting to write to the same file. To fix this, open IIS Manager and ensure the "Maximum Worker Processes" setting for Secret Server's application pool is set to 1. If it is not, set the value to 1 and then either recycle the application pool or perform an `iisreset`.

Unable to complete backup. The following exception occurred: System.Threading.ThreadAbortException: Thread was being aborted

If this error message appears in combination with the application backup files not completed or the size of the file is unusually small, the backup process may have been interrupted by anti-virus software. Disabling scanning of the backup folder should resolve the issue.

Manually Backing up Secret Server

This topic only applies to **Secret Server On-Premises**.

To back up your Secret Server installation:

> Your Secret Server instance may be running during this procedure.

1. Navigate to the directory where Secret Server is installed.
2. Copy the folder (holding the application) to your back up location.
3. Open your SQL Server Management Studio.
4. Right click the database your Secret Server is running on, and select **Tasks > Backup**.
5. Click the **Add** button. You will be prompted to enter a file path.
6. Make sure SQL Server has permissions for this location.
7. Copy the resulting database backup file to your backup location.

> You can also automate steps 2-4 using the command: `osql -S myserver\SQLEXPRESS -E - Q "BACKUP DATABASE SECRETSERVER TO DISK = 'c:\backup\ss.bak'.`

Restoring Secret Server from a Backup

> This topic only applies to **Secret Server On-Premises**.

To restore your Secret Server from a backup:

Restoring the Application

1. Extract your backup zip file of the Secret Server application directory, or copy the files from your other backup location to the physical file path that your virtual directory is pointing to.
2. If you have configured encryption of your `encryption.config` using EFS or DPAPI, you will need to replace the file from the backup with the unencrypted one.
3. Check that FIPS mode is not enabled on the server to avoid an error during the process.

Restoring the SQL Server Database

Choose one of the following scenarios:

Scenario One: Database and Secret Server Are in the Same Location

1. Open SQL Server Management Studio and connect.
2. Right click **Databases** and click the **Restore Database** button.
3. In the **To database** text box, type the database name or select it from the drop down list.
4. Click to select the **Device** radio button.
5. Browse to your database backup file.
6. In the **Restore Database** window Options section, ensure the **Force Restore over Existing Database** check box is checked.

7. Click the **Ok** button.
8. If you get an error saying that Management Studio was unable to get exclusive access to the database:
   a. Right click on the Secret Server database and go to **Properties**.
   b. At the very bottom, change the **Restrict Access** property to "SINGLE_USER". This closes all other connections to the Secret Server database.
   c. Re-attempt the restore.
9. Disable **Force SSL** if there is no certificate installed on the server you are restoring to.
10. In SQL Server Management Studio, expand the databases and select the database for Secret Server.
11. Select **New Query** at on the menu bar to open a query pane.
12. Copy the following command: `UPDATE [dbo].[tbConfiguration] SET ForceHttps = 0` into the query pane
13. Click **Execute** on the menu bar.
14. After the query executes successfully, restart Internet Information Server (IIS) by running `iisreset` from the command line.

> If you are prompted for database credentials when accessing Secret Server and are unable to re-connect, you may need to remap the user.

15. Expand the **Security > Users** folder under the Secret Server database.
16. Remove the user that Secret Server will use to access the database.
17. Expand the **Security > Logins** folder under the SQL Server root.
18. Right click on the log on corresponding to Secret Server and select **User Mappings**.
19. Re-map the log on to the Secret Server database.
20. If necessary, activate your licenses by going to the **Licenses** page.

Scenario Two: The Database and Secret Server Are in Different Locations

1. Delete the `database.config` file from the Secret Server folder.
2. Restart Internet Information Server (IIS) by running `iisreset` from the command line.
3. Use your Web browser to navigate to the new instance of Secret Server. This redirects you to the Web installer because the `database.config` file is missing and it thinks you have not installed yet.
4. Open SQL Server Management Studio and connect.
5. Right click **Databases** and click the **Restore Database** button.
6. In the **To database** text box, type the database name.
7. Click to select the **Device** radio button.
8. Browse to your database backup file.
9. In the Restore Database window options make sure the Force Restore over Existing Database Check box is checked.

10. Click **Ok**.
11. If you get an error saying that Management Studio was unable to get exclusive access to the database:
    a. Right click on the Secret Server database and go to **Properties**.
    b. At the very bottom, change the **Restrict Access** property to "SINGLE_USER". This closes all other connections to the Secret Server database.
    c. Re-attempt the restore.
12. Disable **Force SSL** if there is no certificate installed on the server you are restoring to.
13. Copy the following command: `UPDATE [dbo].[tbConfiguration] SET ForceHttps = 0` into the query pane
14. Click **Execute** on the menu bar.
15. Navigate through the Web installer to Step 3.
16. Type the new database credentials (new server location, username, and password).
17. If you are unable to re-connect you may need to remap the user.

> If you are prompted for database credentials when accessing Secret Server and are unable to re-connect, you may need to remap the user.

18. Expand the **Security > Users** folder under the Secret Server database.
19. Remove the user that Secret Server will use to access the database.
20. Expand the **Security > Logins** folder under the SQL Server root.
21. Right click on the log on corresponding to Secret Server and select **User Mappings**.
22. Re-map the log on to the Secret Server database.
23. Once past Step 3, you are finished. Go to the `home.aspx` page (click the Secret Server logo). There is no need to go any further with the install because the `database.config` has been recreated with the new information.
24. If necessary, activate your licenses by going to the **Licenses** page.

Scheduled Backups

> This topic only applies to **Secret Server On-Premises**.

There are numerous options to consider when backing up Secret Server. Backups can be scheduled to run on a specific time interval. To prevent the directory from growing too large, the number of backups to keep can be defined as well. Depending on size constraints or preferences of the DBA, the database backup can either truncate the transaction log or keep it intact. The additional schedule settings are available when "Enable Schedule Backup" is enabled, and the view page indicates the time and date of the next scheduled backup.

Secret Server Disaster Recovery

The Disaster Recovery (DR) feature is a tool for emergency access to critical systems in the event of an emergency, such as a network outage. DR generates updates at regular intervals from one Secret Server instance (the data source) to another, or many other, instances (the replicas). A select set of folders and the secrets they

contain will have any changes sent securely to the configured replicas so that, in the event the data source becomes inaccessible, the replicas can be quickly used for emergency access to vital systems.

Disaster Recovery and Resilient Secrets

Resilient Secrets Coverage

Overview

The Resilient Secrets (RS) feature is tailored to replicate data from a primary data source to a secondary data replica. The features and specific data replicated **do not** constitute the entirety of an instance of Secret Server. Instead, replicates the prioritized vital data and functionality needed so that, in the event of an outage of the primary data source, the most important information and functionality can be accessed on the secondary data replica to aid restoring the primary data source while keeping minimal operations running.

See Working With Resilient Secrets for more information about using Resilient Secrets with the Delinea Platform.

For more information on the login options available to users depending on service availability and internet connectivity, see Resilient Secrets Login Options.

The following features are what RS replicates, denoting what data is and is not replicated:

Excluded Features

The following features not replicated at all:

- Advanced LDAP Connection Settings
- Advanced Session Recording
- Audits
- Auto Export
- Backup Configuration
- Custom Reports
- Dev Ops Secret Vault
- Disaster Recovery Configuration
- Discovery
- DoubleLock
- Dual Controls
- Event Pipelines
- Event Subscriptions
- HSM Configuration
- Inbox Information and Rules
- IP Address Restrictions
- Jumbox Routes

- Licenses
- Node Information
- RDP Proxy Settings
- Remote Password Changing
- SAML Configuration
- Scripts
- SDK Client Configuration
- Secret Policies
- Session Recordings
- SSH Commands
- Ticket Systems
- Workflows

Conditionally Replicated Features

Configuration

Secret Server

Covered Features

The only global configuration settings replicated are:

- Enable directory service integration.
- Allow authentication against directory services.
- Application hardening state
- Allow quantum state encryption

The SSH cipher suite configuration is replicated.

Exceptions

Any configuration settings not listed above are not replicated.

Delinea Platform

The Platform configuration and permissions are replicated to allow the replica to communicate with Platform when needed, so Platform users can authenticate.

> In the event that everything fails, you will need to have created “break glass” local accounts on the Replica Secret Server. See Resilient Secrets Best Practices for more information. Delinea recommends creating these local accounts as soon as you provision the Replica instance.

For a complete list of login options available depending on service availability and internet connectivity, see Resilient Secrets Login Options.

Folders and Secrets

Covered Features

Folders are replicated, along with extended values associated with their function:

- Subfolders and the entire tree structure, as specified by the RS configuration.
- Any permissions associated with the folder, including whether or not to inherit them from its parent folder.
- For personal folders, the associated user.
- Secret template restrictions placed on folders.

Secrets are replicated along with essential information relating to them:

- The template, active status, name, and the folder the secret is in.
- Several secret settings, including:
  - Multi-factor authentication required
  - Check out enabled or RPC interval
  - Require comment
  - SSH proxy enabled
  - Web launcher requires incognito mode
  - Hide launcher password
- The secret field, item value, and whether or not it is a file attachment. If it is a file attachment, its contents are replicated as well
- Any permissions associated with the secret, including whether to inherit them from the folder.
- Audits are not replicated, but, for data integrity, secret item history, including the history of file attachments, *is* replicated.

Exceptions

These are not covered:

- Password change, heartbeat, and custom expiration information is not replicated because we do not allow RPC to happen from a replica.
- Secret policy information is not replicated.
- Secret dependencies, checkout hooks, and one-time-password settings are not replicated.
- Although TOTP codes are not replicated, you can obtain the MFA code from the source instance and configure TOTP on the replica without any issues.

Launchers and Mappings

Covered Features

All launchers are replicated, along with:

- Their fields.
- Mappings to secret templates.
- Any default associated and privileged secrets.

Exceptions

Custom icons for launchers are not replicated.

Lists

All list information is replicated, including:

- Categories
- Item values
- Team restrictions
- Secret item mappings

Metadata

All metadata on secrets and folders that are configured for replication are replicated. All Metadata on users and groups will be replicated. This includes:

- Metadata field sections.
- Metadata fields.
- Metadata item data.

Roles and Permissions

Roles and permissions are replicated, as are their assignments to users and groups.

Secret Templates

Covered Features

These are covered:

- Secret templates, their settings, permissions, and secret fields are replicated.
- Password requirements, associated rules, and their character sets are replicated.

Exceptions

These are not covered:

- Remote password changing and expiration-related settings are not replicated. Replicas are not intended to run RPC.
- One-time-password settings are not replicated.

Sites

Covered Features

All sites are replicated.

Exceptions

Site connector information is not replicated, and engine information is not replicated. This is because the encryption information related to the site is not replicated.

Teams

All teams are replicated, along with their group and site mappings.

Users and Groups

Covered Features

All users and groups in the system are replicated and, for licensing purposes, we recommend the RS configuration be set to having users inactive by default and only enable the ones needed for vital access. User accessibility is maintained between source and replica. Group memberships are replicated and any associated roles and permissions mapped to them will come across as well.

Any directory services on the source is replicated to the data replica, and all users and groups mapped to these services retain system accessibility, provided the data replica is on a network that can reach the service endpoint.

Exceptions

These are not covered:

- Two-factor-authentication (2FA) configuration or details are not replicated. Users are not automatically configured for 2FA on the replica.
- User passwords are not replicated. Local users need to have their passwords manually reset by an administrator on the replica.

Setup

Secret Server Database Preparation

Overview

We provide several SQL scripts to help you identify any data duplication you may have that could cause issues with DR. We recommend that you run all of them.

These searches will be built into future DR installations. We provide these as an interim measure till that happens.

They are:

- Secret Template Names: Searches for duplicates
- Secret Field Slug Names: Searches for duplicates
- Character Set Names: Searches for duplicates
- Password Requirements Names: Searches for duplicates
- Domain Names: Searches for duplicates of formal domain names (those in URLs)
- Domain Friendly Names: Searches for duplicates of human-readable domain name equivalents
- Folder Paths: Searches for duplicates of secret folders
- Group Names: Searches for duplicates of user group names
- Role Names: Searches for duplicates of user role names

Procedure

Copy and paste each query into a "Creating and Editing Reports" on page 887 and run it. Alternatively, database admins can run the scripts directly on the Secret Server On-Premises database. After running the queries, if you receive any results, that indicates duplicates to address before enabling the DR feature. Most issues can be resolved by simply renaming or removing the duplicate items.

SQL Scripts

Secret Template Names

```
SELECT
    SecretTypeName
FROM
    tbSecretType
GROUP BY
    SecretTypeName
HAVING
    COUNT(*) > 1
```

Secret Field Slug Names

```
SELECT
    st.SecretTypeName,
    t.FieldSlugName
FROM
(
    SELECT
```

```
        sf.SecretTypeId,
        sf.FieldSlugName
    FROM
        tbSecretField AS sf
    JOIN
        tbSecretType AS st2
    ON
        st2.SecretTypeID = sf.SecretTypeID
    GROUP BY
        sf.FieldSlugName, sf.SecretTypeId
    HAVING
        COUNT(*) > 1
) AS t
JOIN
    tbSecretType AS st
ON
    st.SecretTypeID = t.SecretTypeID
```

Character Set Names

```
SELECT
    cs.Name
FROM
    tbCharacterSet AS cs
GROUP BY
    cs.Name
HAVING
    COUNT(*) > 1
```

Password Requirement Names

```
SELECT
    pr.Name
FROM
    tbPasswordRequirement AS pr
GROUP BY
    pr.Name
HAVING
    COUNT(*) > 1
```

Domain Names

```
SELECT
    Domain
FROM
    tbDomain
```

```
GROUP BY
    Domain
HAVING
    COUNT(*) > 1
```

Domain Friendly Names

```
SELECT
    FriendlyName
FROM
    tbDomain
GROUP BY
    FriendlyName
HAVING
    COUNT(*) > 1
```

Folder Paths

```
SELECT
    FolderPath
FROM
    tbFolder
WHERE
    UserID IS NULL
GROUP BY
    FolderPath
HAVING
    COUNT(*) > 1
```

Group Names

```
SELECT
    GroupName
FROM
    tbGroup
WHERE
    IsPersonal = 0
    AND IsPlatform = 0
    AND SystemGroup = 0
    AND DomainId IS NULL
GROUP BY
    GroupName, DomainId
HAVING
    COUNT(*) > 1
```

Role Names

```
SELECT
    r.Name
FROM
    tbRole AS r
WHERE
    r.RoleType = 1
    AND r.IsSystem = 0
GROUP BY
    r.Name
HAVING
    COUNT(*) > 1
```

Duplicate Launcher Type Fields

Duplicate launcher type fields in the source database can cause a DR sync failure. This script detects them.

```
SELECT
  ltf.Name AS LauncherTypeField,
  lt.ConcurrencyId AS LauncherTypeConcurrencyId
FROM
  tbLauncherTypeField ltf WITH (NOLOCK)
  INNER JOIN tbLauncherType lt WITH (NOLOCK) ON lt.LauncherTypeId = ltf.LauncherTypeId
GROUP BY
  lt.ConcurrencyId,
  ltf.Name
HAVING
  COUNT(*) > 1
```

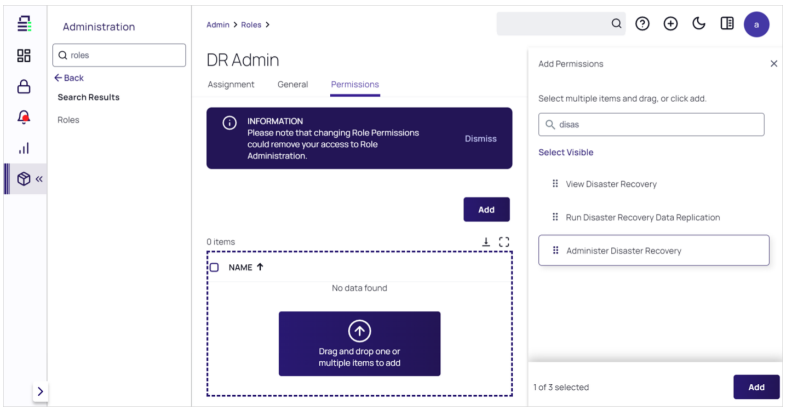
It is possible to use Disaster Recovery with Integrated Windows Authentication enabled on Secret Server Cloud. See "Configuring Integrated Windows Authentication" on page 383 for more information.

Role Assignment

Assign the "administer disaster recovery" role permission:

Do this on both the data source and replica instances.

1. Click **Administration** in the main menu. The Secrets Administration page appears.
2. Click the **Roles** link in the **Users, Roles, Access Management** section. The Roles page appears.
3. Select or create the role you want to configure this feature with.
4. Click the **Permissions** tab.



5. Click the **Add** or **Edit** button.
6. If you do not see the role in the **Add Permissions** list, type `Administer Disaster Recovery` in the search box.
7. Drag the **Administer Disaster Recovery** role from the **Add Permissions** list and drop it in the unlabeled permissions box. The role now has the permission.

Data Source DR Configuration

Configure the data source DR configuration:

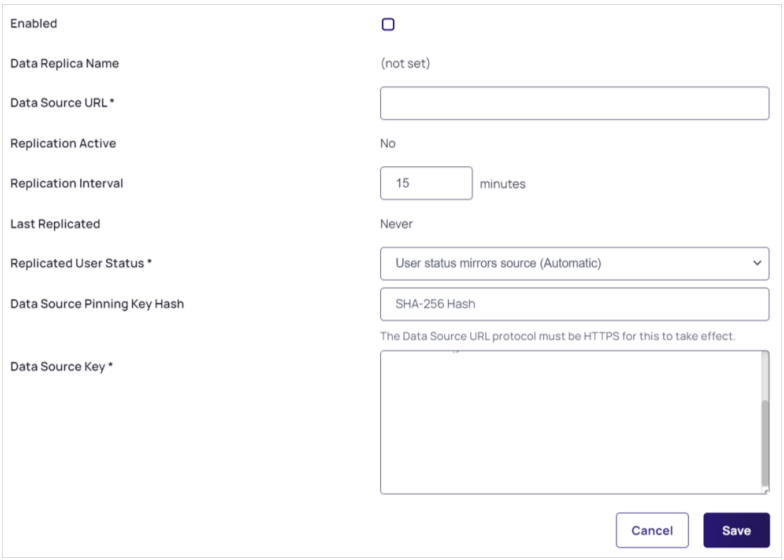
1. Click **Administration** in the main menu. The Secrets Administration page appears.
2. Search for and click **Disaster Recovery**. The Disaster Recovery page appears.
3. Click the **Outgoing Configuration** tab.
4. Click the **Outgoing Setup Steps** button. A popup appears.
5. Click the copy button to copy the URL from the **Data Source URL** and save it where you can easily access it.
6. Click the **Copy Data Source Key** button to copy the key to the clipboard. The popup disappears.
7. Copy the key and save it to the same location you used for the URL. It should look something like this:

```
https://mydomain.com/Playground
                                        BgIAAACkAABSU0ExABAAAAEAAQAPeEYJLZ3u1F26EF+bBiRwokrGusSAICUpR
2/lo3Ad7dpTEl7j3rvG9+T+j8DKTsmi6Xfj
c/J0hkoX00b6LOq4feo6mvnf6Lp8agopN2XWjLy4KU7ICG2iAoL4wgIdpgWVCHdZUcHH
Vhe6RSkahRDOC4ctpRxb/KOI8Bbk6ftporjwZAfEfCVEOotnnm4Z8qjI0XKKbhLl2eTH
2lDTJk53dt8z7g/Aj4nz7aYUMJX1vDOKKARpd3GiVfu5fcb0hpIqw64pjOs8trqOVxuy
oyfSLwYPC9rtK4/JZ7Xodq93p9IIEJfSY7cup9kCHhFzU8d3RNgWj5pKKVLsPOVUreXV
6vceGmRA3OeKY/a8/I6aEAkmkWjJAP5bmMosYVbcRR2PxI0bxXGUSKbMdKdC2Akgm3PW
jWqezONsv0QFmJPYEGrLIxeJc2/9X+TUnx5ON/Cgeb2bHSf5CLXizjxlB67gzDZpH3q0
mmJzm0aSDmqIc20UxFYk2YZeEzICI5lnAVPiAbaCoKuhQdhaxxaD+wdMQDCCEooKiyhW
Yo3NFYY79k6scJh2+sn62xZNPdpxi0m6rP/F+mpELKMyElwJdFxVm+l14ksaKhIA6bDn
xRLJ/f/SMdxfmIcTqxLr/r9k6RAiJo8DiBoWMkjk4VUCFIL4rYpUIJb/aKmx9npg==
```

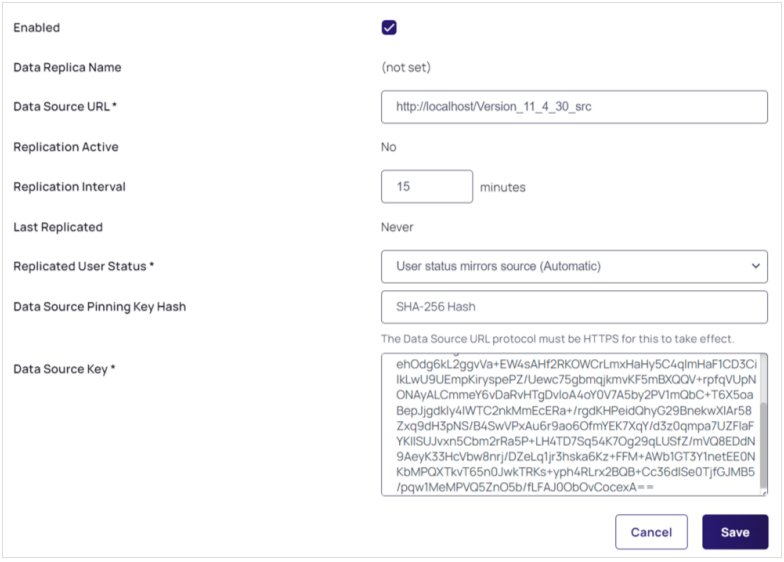
> If you have a On-Premises source instance with multiple webnodes, you will need to enter the storage path where files related to data replication will be temporarily stored. All servers must have a shared path to store the replication packages, otherwise replication errors may occur.

Replica DR Configuration

1. Click **Administration** in the main menu. The Secrets Administration page appears.
2. Search for and click **Disaster Recovery**. The Disaster Recovery page appears.
3. If necessary, click the **Incoming Configuration** tab.
4. Click the **Edit** button. The page becomes editable.

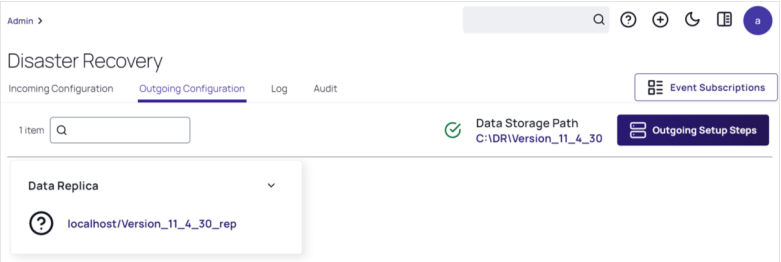


5. Click to select the **Enabled** check box.
6. Type or paste the data source URL you saved before.
7. Type or paste the data source key you saved before.

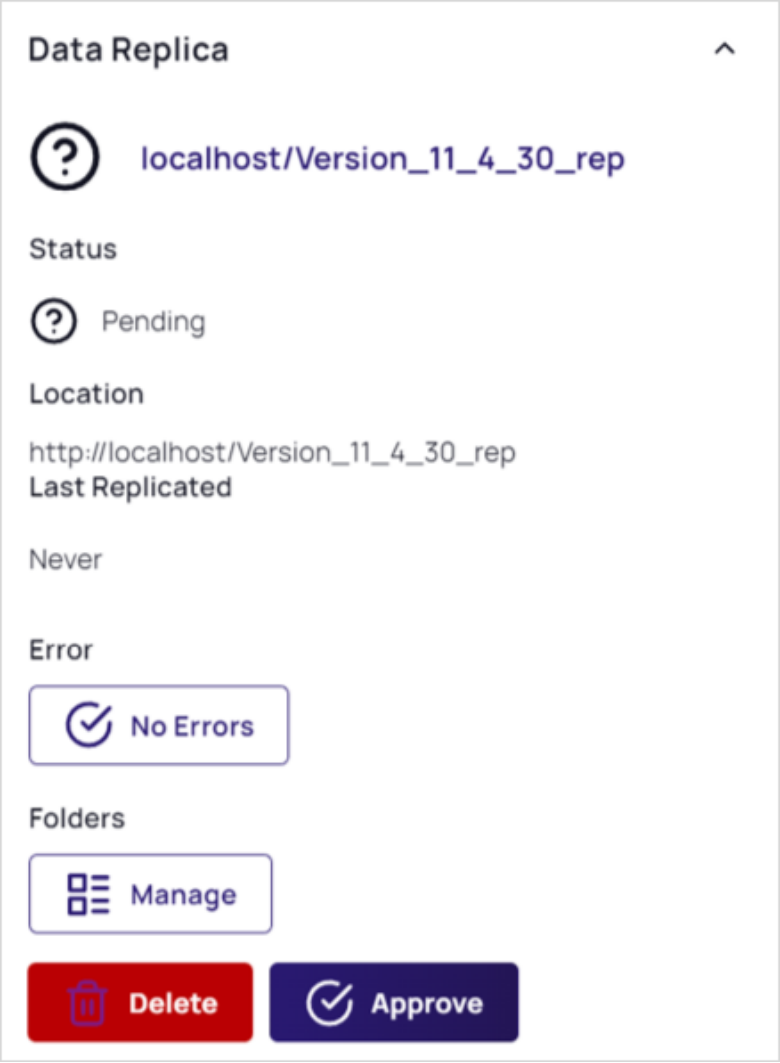


8. Click the **Save** button.

9. Refresh the **Disaster Recovery** page showing the **Outgoing Configuration** tab on the Data Source Disaster Recovery page. A new block appears for the new replica connection:



10. Click the **v** on the block to expand it.

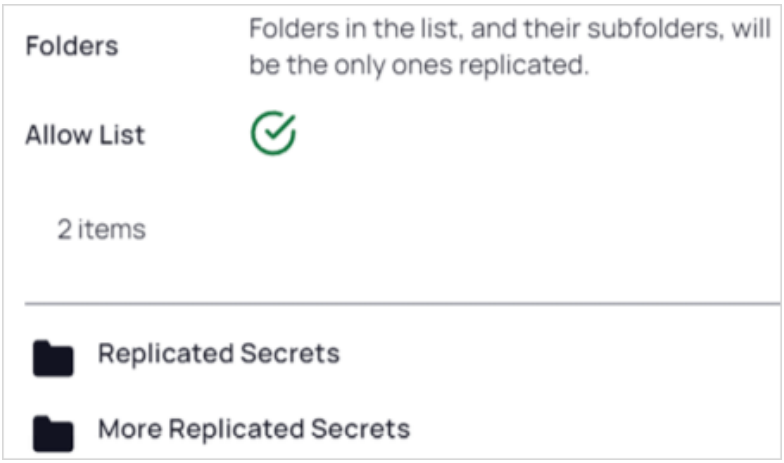


11. Click the **Approve** button.

12. Returning to the **Ingoing Configuration** tab, click the **Test Connection** button to verify everything is configured correctly.

Finishing up

1. Once configuration is complete, return to and refresh the **Outgoing Configuration** tab.
2. We recommend expanding the block and clicking the **Manage** button for folders.



This enables you to select which folders, subfolders, and secrets get replicated. This reduces the scope of the replicated data, enabling faster replication and a smaller duplicated data footprint.

This will enable the regular interval synchronization

Replication

By design, the first time a replication from the data source to the replica occurs, all data is replicated. This is likely the slowest DR operates and can take up to 20 minutes to replicate a data source containing 100,000 secrets.

All subsequent replications are significantly smaller. Only information that has been changed since the previous replication is replicated again.

Replica Mode

When you configure a Secret Server instance as a replica, it is automatically placed in replica mode. While in replica mode, the following features are disabled

- ConnectWise synchronization
- Discovery
- DoubleLock
- Heartbeat
- Pipeline and bulk operations on secrets
- Remote password changing

- Secret import
- Secret policy changes to secrets

Replicated User Status

The "replicated user status" setting determines how the "enabled" status for users is handled during replication. The setting is used primarily when the source and replica user counts do not match–that is, new users are on the source that are not on the replica. Delinea strongly recommends both source and replica user counts match so all users and their access is replicated to the DR instance.

> When these settings are changed, they impact any new users from that point forward. Existing user statuses do not change.

The possible settings are:

- User status mirrors source (automatic): This is the default setting. Any new users in the source are replicated as enabled.
- New users from source are disabled by default (manual): This setting is for admin special cases and must be manually set. New users are replicated as disabled.

Recommendations

We recommend placing a Secret Server replica into maintenance mode. This further reduces the possibility of data changes to the instance. See Maintenance Mode section for more details.

Disaster Recovery Best Practices

Delinea offers robust disaster recovery capabilities in Secret Server, which includes:

- High Availability and resiliency through regional failovers, globally distributed data centers, web server clustering, database mirroring and secrets resiliency.
- Enhanced disaster recovery features such as automated redundancy, seamless failover, and hybrid failover for both on-premises and cloud deployments.

General Best Practices for Disaster Recovery

- Backup Data with Resilient Instances - Resilient instances refer to the capability of replicating secrets data to another cloud or on-premises instance of Secret Server with automated one-way synchronization from the source instance to the replica. Replicas should be kept in read-only mode to avoid loss of integrity. This ensures continuous access to secrets, even during emergencies, thereby reducing the risk of downtime or disruption in privileged access. See Working with Resilient Secrets to learn more.
- Using Multifactor Authentication (MFA) - Secret Server offers robust Multi-Factor Authentication (MFA) capabilities to enhance the security of accessing privileged accounts and sensitive information. Key aspects and resources related to MFA in Secret Server include MFA enforcement of credentials, MFA on secrets for

Secret Server Cloud customers, integration with Microsoft Authenticator and a lot more. Refer to "Multi-Factor Authentication" on page 433 for more information.

- Security Hardening - Security hardening for Secret Server involves implementing a series of best practices and configurations to enhance the security of your Secret Server instance. This includes securing the operating system, application settings, database, and network communications. Refer to the "Security Hardening Guide" on page 1402 for more information.

> Please be sure to test the items in the Security Hardening Guide one at a time as multiple simultaneous changes can cause issues in case you need to test or revert the changes.

- Setting Up Event Alerting - provides robust alerting and notification features to help administrators stay informed about critical events and actions. Refer to "Event Subscription Overview" on page 310 for information.
- Leverage Rabbit MQ's Disaster Recovery Capabilities - The Best high availability/disaster recovery multi-site deployment for RabbitMQ Helper in Secret Server is designed to provide high availability and disaster recovery across multiple locations, typically a primary and a secondary disaster recover site. This setup ensures that RabbitMQ Helper clusters are available in multiple locations, providing robust failover capabilities. Learn more about Rabbit MQ.

Server Clustering

> This topic only applies to **Secret Server On-Premises**.

Secret Server can run with multiple front-end Web servers. For a critical instance, clustering offers a redundant system to limit potential down time from a single point of failure. Clustering also allows users to load balance for better performance. For instructions on enabling clustering in Secret Server, see "Secret Server Clustering" on page 776.

SQL Server Mirroring

> This topic only applies to **Secret Server On-Premises**.

This topic describes the process of configuring Secret Server and SQL Server for a high-availability environment using Mirroring. The contents of this paper include:

- Configuring SQL Server 2016 for database mirroring with a failover partner and a witness
- The encryption used between the primary database and the mirror database
- Configuring Secret Server to use mirroring to achieve high availability

> This topic uses SQL Server 2016, but it is very similar to earlier versions.

SQL Server Mirroring is deprecated as of SQL Server 2016 and is not supported in SQL Server 2022. For environments using SQL Server 2022, it is recommended to use Always On Availability Groups (AAG) instead. If the SQL Server edition does not support full AAG, consider using Basic Availability Groups as an alternative. For more information on configuring Basic Availability Groups, refer to the Microsoft documentation on Basic Availability Groups.

Introduction

Three different SQL Server instances are required to implement this scenario:

- **Primary database:** The main application database
- **Mirror database:** Replicates all of the data on the primary database in a transactional manner
- **Witness database:** Monitors the health of the primary and mirror databases and initiates failover if necessary

In the setup described here, mirroring operates in synchronous mode, which means that a transaction does not commit on the primary database until it has committed on the mirror.

See Prerequisites, Restrictions, and Recommendations for Database Mirroring for more on synchronous mirroring.

Procedures

Setting up Databases for Mirroring

To initiate database mirroring, the databases on the primary and secondary machines must have the same name. We recommend doing this before installation. To initially set up mirroring, in Microsoft SQL Server Management Studio, take a full backup of the database on the primary and then restore it onto the database on the secondary. When restoring the database, the "RESTORE WITH NORECOVERY" option must be selected.

SQL Server Configuration

The three SQL Server instances should all be running under the same domain account. It is possible to run under different accounts but the configuration is more complex and not supported by Delinea technical support. Each SQL Server instance should be configured to listen on TCP.

Configuring Mirroring

To configure mirroring:

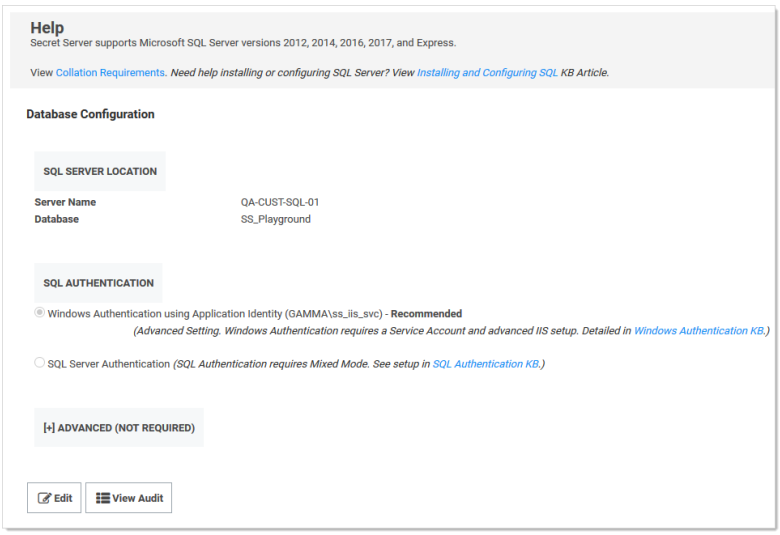
1. In Microsoft SQL Server Management Studio, drill down to the primary database in the Object Explorer.
2. Right click the primary database and select **Properties**. The Database Properties window appears.
3. Select the **Mirror** page.
4. Click on the **Configure Security** button. The Configure Database Mirroring Security Wizard appears on the introduction page.
5. Click the **Next** button. The Include Witness Server page appears.

6. Click to select the **Yes** selection button.
7. Click the **Next** button. The Choose Server to Configure page appears.
8. Click to select all three interface check boxes (principal, mirror, and witness servers).
9. Click the **Next** button. The Principal Server Instance page appears.
10. Click the **Principal server instance** dropdown list to select the current (primary) server.
11. Type a port number for connecting to the other servers in the **Listener port** text box. The port must be open for TCP communication on the machine's firewall and on any network devices that restrict access to this machine.
12. Click to select the **Encrypt data sent through this endpoint** check box. This enables RC4 encryption on data sent through this endpoint.
13. Type `Mirroring` in the **Endpoint name** text box. The endpoint name is for referencing the endpoint later.
14. Click the **Next** button. The Mirror Server Instance page appears.
15. Repeat the exact same configuration you set for the primary server instance with only the server instance name different (choose the mirror instance).
16. Click the **Next** button. The Witness Server Instance page appears.
17. Repeat the exact same configuration you set for the primary server instance with only the server instance name different (choose the witness instance).
18. Click the **Next** button. The Service Accounts page appears.
19. Type the domain user that SQL Server runs under for each instance's Service Accounts text box. For example `mydomain\sql_svc`.
20. Click the **Finish >>** button. Logins are created for each account and are given CONNECT permission on each endpoint, if needed. The Complete the Wizard page appears.
21. Click the **Finish** button

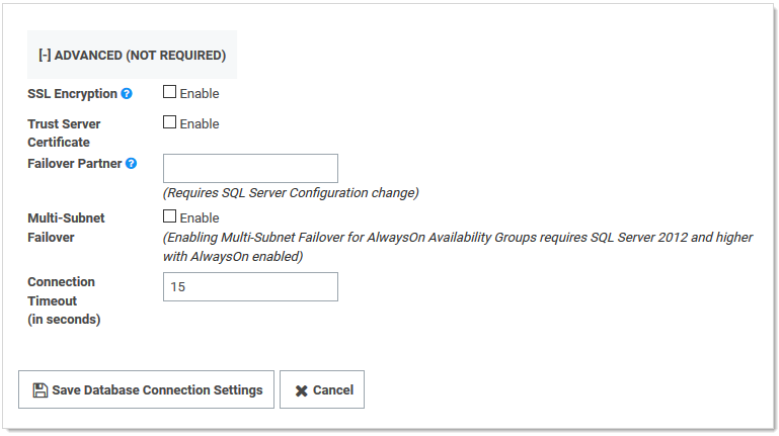
Configuring Secret Server for Mirroring

> The credentials used to access the primary database must also be valid on the mirror database for failover to work.

1. Go to **Admin > See All**. The admin panel appears.
2. Type Database in the **Search** text box and select **Database**. The Database Configuration page appears:



3. Click the **Edit** button.
4. Click the **Advanced (Not Required)** link. A new section appears:



5. Click the select the **SSL Encryption** check box.
6. Type the mirror server name in the **Failover Partner** text box.
7. Click the **Save Database Connection Settings** button.

Testing Mirroring

This procedure is necessary to verify that failover will function correctly in the event that the primary server is unavailable or inoperable:

1. Open SQL Server Enterprise Manager.
2. Right click the primary database and select **Properties**.
3. Click the **Mirroring** tab.

4. Click the **Failover Now** button. This causes the database on primary to switch roles and become the mirror database. The mirror database becomes the primary. Clients using the application should be able to continue as before.

> One request may fail before Secret Server begins making requests to the new primary database.

Database SSL Configuration

> See Enable encrypted connections to the Database Engine for instruction on configuring SSL for SQL Server.

The certificate authority used for the SSL certificates must be trusted on all of the machines that are a part of Secret Server's installation. The SQL Server service account must be granted access to the certificate.

Procedure:

1. Open Microsoft Management Console by running `mmc` on the Windows command prompt.
2. Drill down to **Console Root > Certificates > Personal > Certificates** in the navigation tree.
3. Right click the certificate and select **All Tasks > Manage Private Keys**.
4. Grant the user account that SQL Server uses read permission.
5. Ensure SSL is enabled for both the primary and mirror database server. See Configuring Secret Server for Mirroring. It is not necessary to configure SSL on the witness server.

SQL Server Replication Best Practices

> This topic only applies to **Secret Server On-Premises**.

> **Geo replication is deprecated. It is no longer a feature of Secret Server and is not supported by Delinea.** When it was active, Geo Replication upgrades were outside the scope of technical support and were scheduled through Professional Services. Due to improvements in AlwaysOn technology, we recommend using it instead for most use cases.

Overview

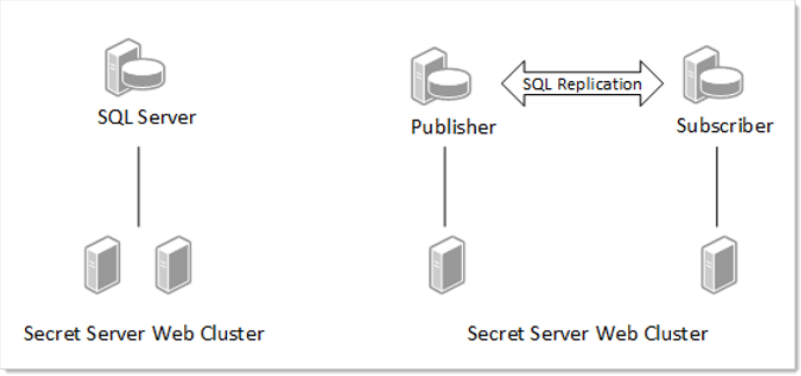
Secret Server geo replication is SQL Server replication using a set of technologies for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency across geographically distributed data centers.

> Secret Server uses merge replication. The only version validated is pull-based merge replication. Push based replication has not been validated.

SQL timeouts are a known error condition when SQL Server Always On is configured to use synchronous replication. We recommend using asynchronous replication to mitigate or avoid the timeouts. Secret Server does hundreds of transactions per minute from all web nodes, any of these SQL operations can timeout if the replication takes too long blocking other operations to complete in a timely manner. Asynchronous replication will not block other operations from completing in a timely manner.

SQL Server Replication

Figure: SQL Server Replication



Benefits of Replication

Enabling SQL Server replication allows a database and application to be hosted closer together and this allows for the mitigation of network latency and outages.

- Decrease application server to database network latency
- Resolve issues with unreliable network connectivity
- Allows for distribution of workload in a scale out fashion
- Works across large distances

In a typical web-clustered version of Secret Server, all application servers access a centralized database. In the event of a network outage, any users on affected application servers are not be able to use Secret Server until network access was restored. In addition, poorly performing networks can introduce latency that may decrease the responsiveness of Secret Server. This technology provides additional options when designing the network topology behind Secret Server that can help alleviate these issues.

High Availability

SQL Server replication is *not* an option for high availability, but it can be coupled with other technologies like SQL Server AlwaysOn Availability Groups to provide high availability. Any architecture should be reviewed and designed with your database group.

Architecture

The SQL Server Replication technologies do all the work of ensuring data consistency between each database, and Secret Server is designed to work well with this technology. When SQL Server replication is enabled on a specified database several system tables, views, stored procedures, and SQL Server jobs are added to the database schema. These tables store information about the data replication, and the procedures contain most of the code needed to perform the synchronization between databases.

Data Synchronization

This is the process through which SQL Server replication integrates changes from each database node. These changes include data changes as well as schema changes. Each subscribing database node has a synchronization interval that defines when it will synchronize any changes between the main publication node and itself. Secret Server has been tested using pull-based subscriptions where a scheduled job on each SQL Server subscriber runs at a specified interval to trigger the synchronization.

Data Conflicts

Enabling SQL Server replication introduces the possibility of data conflicts occurring in the Secret Server environment. This can happen when two people in different regions attempt to update the same set of data. Due to the disconnected nature of the technology, the system is unaware of this conflict until a data synchronization occurs. During synchronization, SQL Server attempts to resolve any conflicts based on a defined set of parameters. Secret Server provides a setup script to help define optimal parameters for each article (table, view, and stored procedure) in the Secret Server database.

SQL Server Replication Monitor and Conflict Viewer

For more information:

- Replication Monitor
- Conflict Viewer and Interactive Resolver

Tracking level

SQL Server replication tracks changes to each node by either row or column:

For more information on row- and column-level tracking, see Row-Level Security

Conflict Resolvers

SQL Server replication uses resolvers to determine the outcome of data conflicts between two database nodes. For example, the same row or column was updated on different nodes. These can be as simple as the publisher database always wins, last change by date wins, and many others.

Switch nodes over to subscriber databases. Any nodes that are configured to run background, engine, or session recording roles must remain on the publisher database.

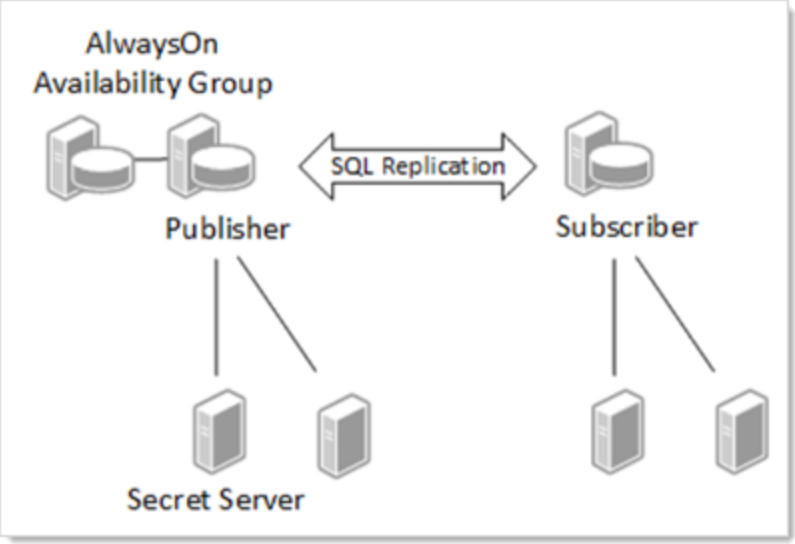
For more information see:

- Advanced Merge Replication Conflict Detection and Resolution
- Specify Merge Replication Properties

Secret Server and SQL Replication

There are many architectures for how SQL Replication can be setup. Determining the correct configuration requires proper planning with a good understanding of how SQL Replication works and the intended goals of using this technology. Here are a few examples.

Figure: Web Cluster with SQL Server Replication



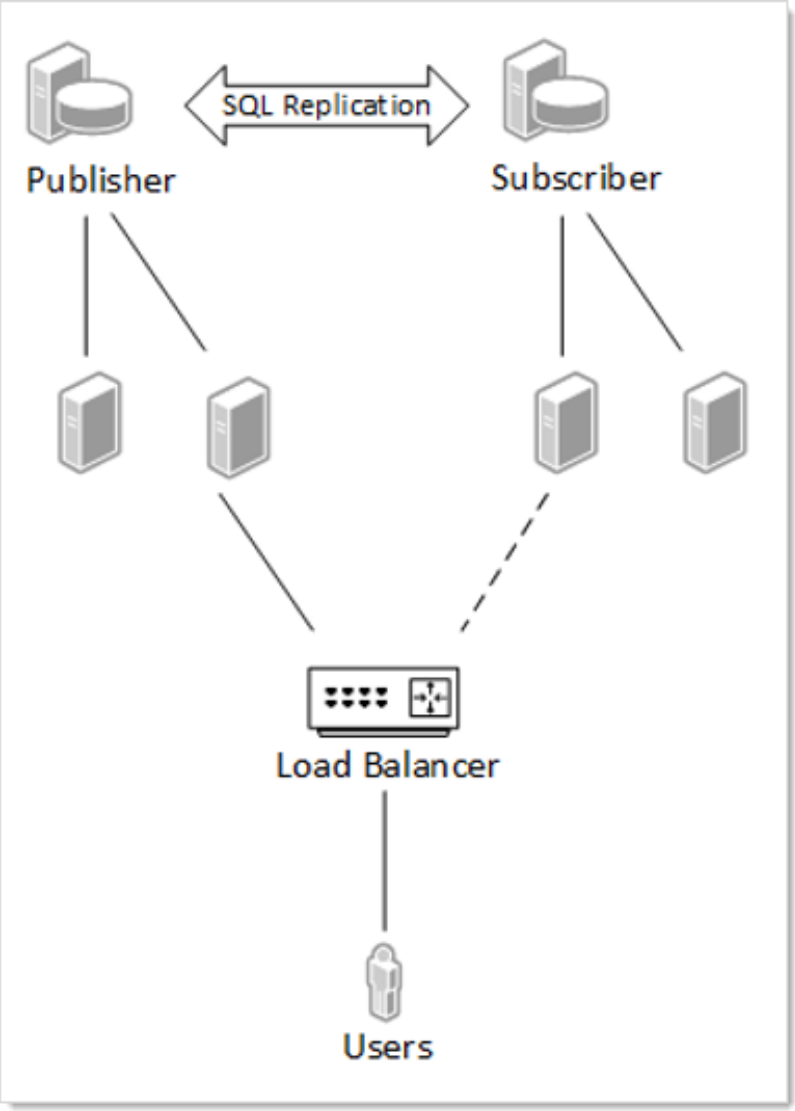
Some key points about designing the right architecture:

- Multiple web application servers can connect to the same database server.
- Secret Server allows you to configure various roles (background, engine, session recording) that perform various services, such as heartbeat and password changing. These roles can only function on a node that has a connection to the publisher database and will not run on other nodes even if configured to do so. If no suitable node is available that has roles enabled and is connected to the publisher database, then certain activities such as secret heartbeat and remote password changing will be offline until such a node becomes available.
- Secret Server Engines are an effective means to distribute workload to different networks or sites. Each engine must call back to a Web application node that connects to a publisher database.
- The diagram shows the publisher in an AlwaysOn availability group, but this is just an example. Depending on the needs of the organization, AlwaysOn could run on both the publisher and the subscribers or on neither. Many other high availability options could be leveraged alongside SQL

  Replication.

Using a Subscriber When the Publisher Is Offline

Figure: Using a Subscriber When the Publisher Is Offline

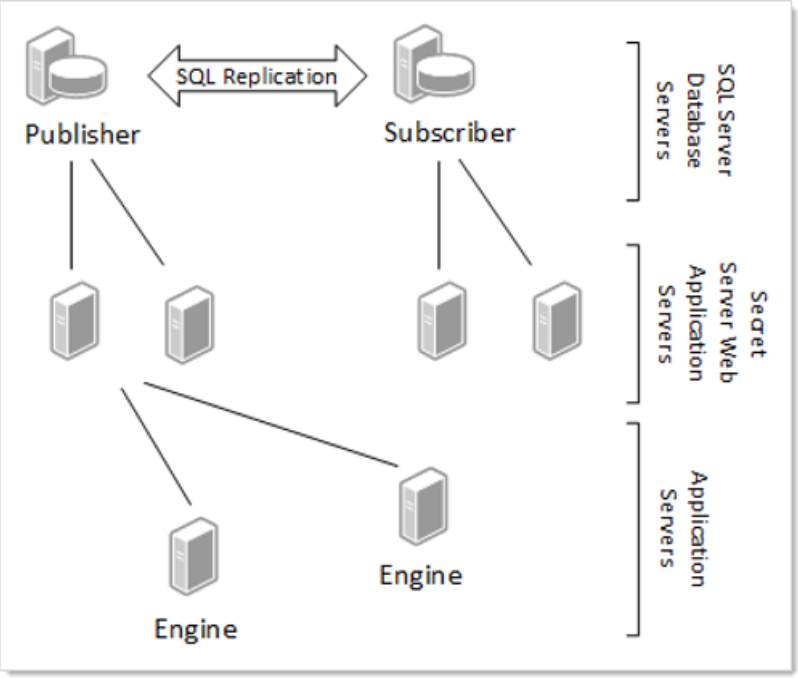


In the event of a network outage, a load balancer could be configured to fail over to the subscriber nodes. The data on the subscriber would be as up to date as the last synchronization, and when network connectivity is restored all data activity will be synchronized again. Obviously, if the issues of network latency or disconnects occur too frequently, this may not be a workable solution.

Secret Server Distributed Engine

You can add Secret Server distributed engines to different network segments, but you must still be able to call back to a Secret Server Web application server that is connected to a publisher database node. This helps to ensure that heartbeats and password changes occur with the most up to date dataset.

Figure: SQL Replication and Distributed Engines



Secret Server Replication Settings

Keep in mind that this is a disconnected technology, so there are required decisions when setting it up:

- How often should synchronization occur?
- Should any resolvers override the provided defaults?
- Are there any operational considerations based on the type of secret data?

Publications

- Events: These are things that need to occur on a timely schedule, such as updating secrets. The events publication should be set to synchronize every minute. This means that if a secret were created in one region, it would not appear in another region for one minute. By default, this publication is called *SSPubEvents*.
- Logs: Information from each audit log is available on its server. We do not recommend setting log synchronization to less than hour. If this happens too frequently, the database could create deadlocks by constantly updating large sets of data. Once the logs merge, the events will appear in the order in which they occur and show on the server on which they took place. By default, this publication is called *SSPubLogs*.

Using the default Secret Server implementation, most conflicts are resolved by taking the change made on the publication server as the winner (using the "publisher wins" resolver). It is assumed that the publication server is most likely the main server in an environment and therefore, most likely, the decider in a case of a conflict. Thus, we recommend doing functions, such as configuration changes or secret template definition changes, on the publication node. This is only an issue if two people update the same data on two different servers before a synchronization occurs.

There are some exceptions to the "publisher always wins" rule:

Data Element	Resolver or Tracking Level	
Secret Fields	Last Update Wins	If two people change a password on a secret to a new password, then the last person to make the change will win.
Secret Access Request Approval	Last Update Wins	If two people are in a group that is requested to approve access, then the last person to approve or disapprove access would set that approval.
File Attachment	Last Update Wins	
tbConfiguration	Column Level	Configuration changes are resolved by table column, not the default row-level where the entire row is resolved. This allows different configuration options to be set on different servers and not conflict with each other, but changing the same option would create a conflict.

Always remember that the conflict is only resolved when synchronization occurs. If there is a synchronization window of 24 hours, there could be very different data on different servers for that entire 24-hour period. The risk of data conflicts increases with larger synchronization windows. The default recommended synchronization is every minute for most items and every hour for system logs.

For a complete listing of recommended settings navigate to the **SQL Replication Administration** page in an installed version of Secret Server at **ADMIN > Nodes > SQL Server Replication > Show Articles**.

Conflict Auditing

When conflicts occur during synchronization, Secret Server writes an event to the system log on the server that it has detected these conflicts. Depending on the type of conflict, more-specific information maybe written to the audit log for a specific data entity, such as a secret or folder. Notification of the conflict is written to the system, secret audit, folder audit, and other logs. To view complete details of the conflict, the SQL Conflict Viewer tool needs to be used to review the conflict by right clicking on the publication and choosing to View Conflicts.

As an example, if the same field changed on two different servers, resulting in a conflict, then the Secret Server audit log will indicate the field was changed twice and then denote that there was a data conflict.

Operational Latency

Give consideration to how your Secret Server data is setup and managed. This is most easily described by an example where a manager uses a Secret Server node in Europe and their employee works on a node in Australia. Due to network latency, the organization has changed the default event log synchronization interval from every minute to every five minutes. The employee requests access to a secret in Australia, but due to the synchronization window, the manager is not alerted until five minutes later because the manager is using the server in Europe. You can mitigate this issue by having operational groups work on data for their region or decreasing the synchronization window.

Options to manage data latency if data conflicts occur regularly:

When setting up the SQL Replication Publication certain articles need settings other than the default settings. The recommended settings can be found within Secret Server by accessing the SQL Server Replication page located at

Admin > Nodes > SQL Server Replication > Show Articles. This page can also be used to generate a setup script for both the Publication and Subscribers that uses these default settings. A Distributor will need to be created before running these scripts. For more information, see Configuring Distribution.

Compensate for Errors

When conflicts occur, an article that has the compensate_for_errors attribute set to true will automatically try to resolve the conflict. When false, a SQL Server administrator can use the SQL Server Conflict Viewer to review and resolve conflicts.

Identity Range

SQL Server replication manages table identities by assigning ranges to the publisher and subscribers. Certain tables (logging or auditing) require larger assigned identity ranges. New ranges are only assigned when a data synchronization occurs. For more information, see Replicate Identity Columns.

Variations

How SQL Server replication is setup can vary greatly and there may be reasons to not use the standard setup. Consult with your database group for approaches that may work well in your environment. The architecture diagrams contained within this document are just high-level examples.

Installation

There are a multitude of configuration options for SQL Server replication. At a high level, these are the steps to setup Secret Server in a SQL Server replicated environment:

> This can run on the same database as the publication database or on a separate one on another server.

1.
2. Review what settings are appropriate for the distributor setup with your on-premise database group or with a database consultant.
3. Download the SQL publication script.
   a. In Secret Server select **ADMIN > Nodes > SQL Server Replication >Get SQL Publication Script**.
   b. Review this script and update the variables according to your environment.

> Advanced users can use the article list on that same page to configure SQL replication differently than this script to suit your environment.

1. Run this script on the Secret Server Database. A DBA with administrative privileges is needed to run this. Please consult with your on-premise database group or review your configuration with a database consultant.
2. Create snapshots for each publication:

   a. Open SQL Server Management Studio
   b. Right click on each publication under **Replication > Local Publications**
   c. Select **View Snapshot Agent Status**.
   d. Click **Start**
3. Download the SQL subscriber script:
   a. In Secret Server select **ADMIN > Nodes > SQL Server Replication >Get SQL Subscriber Script**.
   b. Review this script and update the variables to match your environment
4. Create a new database on the database server you intend to be your subscriber. Ensure the script uses this database and machine name. Set up permissions for the user or network account that Secret Server uses to connect to this database.
5. Run the subscriber script on the publication database first and then on the subscriber database. If your variables are set properly, it will execute the appropriate part of the script.
6. Expand the **SQL Server Jobs** on the subscriber, and you should see two jobs named for each publication.
7. Right click these jobs to start them. After they complete, your subscriber database should have replicated the schema objects from the publication.
8. Switch Secret Server nodes over to subscriber databases. The primary node *must* remain on the Publisher database, as must any node that an engine calls back to, but all other nodes can be reconfigured to use subscriber databases. Which nodes to switch depends on your specific needs as described in the previous sections. To switch an existing node to a subscriber database, log into that node and go the DbConnectionReset.aspx page by entering that page name in the URL field of your browser (`http[s]://<your_secret_server_name>/DbConnectionReset.aspx`). Step through the wizard, entering the name of the new server and database when prompted. After completing this step, recycle the node's application pool.

Troubleshooting the Installation

Replication Setup Scripts Fail

Make sure that the SQL Server replication feature is enabled before running the script. Check the error messages in the script output. Make sure you set all of the variables in the top of the script correctly (such as publisher server, database names, and subscriber server).

SQL Replication Job Fails

To see the error message, it is easiest to right click on the job and choose to view history. This error message can indicate the actual problem.

Removing SQL Server Replication

Certain operations such as upgrading Secret Server, adding or removing DoubleLocks from secrets, and enabling or disabling HSM cannot be performed while SQL Server replication is enabled. In order to perform these operations you must remove SQL Server replication. When you are done with the action that required you to remove replication you can install and configure it again by repeating the previous instructions.

You only need to remove replication from the Publisher and all Subscribers. The Replication Distributer does not need to be removed. These are the steps to remove SQL Server replication from your publisher and subscriber databases.

On Each Subscriber

1. Stop the websites of all nodes using the subscriber database.

> You can see what database each node is using from **Admin > Server Nodes**.

1. In SQL Server Management Studio, go to , right-click **Local Subscriptions** and choose
2. In the dialog, click to select "Subscriptions in the following data sources" and select the Secret Server subscription databases.
3.
4. Click **Generate Script > Open in new query window**.
5.
6. The script contains sections to be run on both the subscriber and the publisher. Run the sections on the subscriber by uncommenting them and commenting those for the publisher.
7. Copy the script to a query window on the publisher server.
8. The script contains sections to be run on both the subscriber and the publisher. Run the sections on the publisher by uncommenting them and commenting those for the subscriber.
9. Perform any maintenance actions needed on the subscriber database and nodes.

On the Publisher

1. Stop the websites of all nodes using the publisher database.
2. On the publisher, go to **Replication**, right-click **Local Publications** and choose **Generate Scripts...**
3. In the **Generate SQL Script** dialog **check Publications in thefollowing data sources** and select the Secret Server database.
4. Click to deselect the **Distributor Properties** check box.
5. Select **To drop or disable the components**.
6.
7. Click **Close** to close the dialog once the script is created.
8. Run the script.
9. Perform any maintenance actions needed on the subscriber database and nodes.
10. After all maintenance tasks are done, restore replication as described in Installing and Configuring SQL Server Replication.

Managing SQL Server Replication

Once replication is setup and working certain considerations should be given to managing it along with conflicts that can occur. The recommended settings for the publication have been tested to limit conflicts, but they can still occur. Here are some scenarios you might encounter:

Scenario	Solution
Conflict automatically resolved	SQL Server was able to determine how to resolve the conflict. Secret Server will log and audit this conflict. To see the specific details for the conflict, use the **SQL Server Conflict Viewer** in SQL Server Management Studio.
Conflict unable to be automatically resolved	A user with SQL Server access needs to open the conflict in SQL Server Management Studio. Access these by right-clicking on the publication and choosing to view conflicts.
Some data stops replicating	A table could become blocked if there are conflicts. Other data may continue to synchronize. If data in one region or database node is different than another after a synchronization, there could be conflicts that need to be reviewed and resolved.
SQL Replication Synchronization Times	The status of each publication and subscribing server along with the last time of synchronization can be located in Secret Server by selecting .

Web Server Nodes

The **Web Server Nodes** page now includes a column that lists the database server name and database name. This column also indicates whether the database is the publisher or a subscriber. To see the page:

1. Open Secret Server.
2. Click the **Admin** menu item and select **All**.
3. Click the **Server Nodes** button. The Server Nodes page appears (not shown).
4. If you click the **SQL Server Replication** button, you can see more information about your SQL Server replication. The page pulls from the replication data and shows the:
   - Publication name
   - SQL server
   - Database
   - Subscription type (push or pull)
   - Status of the last sync
   - Last time that subscription was synced
   - Date of the last sync

The same information is available within SQL Server Management Studio, but this page gathers all of the subscription information together in one place.

As mentioned earlier, the **Get SQL Publication Script** and **Get SQL Subscriber Script** buttons will download replication script templates that you can use to set up replication for Secret Server. You can fill in the variables at the top of each script to match your environment and run them as-is or modify them further if you need to customize the default scripts.

The **Show Articles** button lists each article in Secret Server that should be included in replication along with the recommended settings for SQL Server replication. These are *recommended settings* and do not show the current state of replication on the publisher.

Upgrade Scenario

New versions of Secret Server may issue schema changes including indexes, column changes, and views. In some cases, SQL merge replication will not automatically replicate these schema changes. For this reason, we recommend removing any publications and subscriptions targeting the Secret Server database, redirecting users to the web server at the primary site before performing any upgrade, and recreating the publication and subscriptions from new versions of the replication scripts.

In the following scenario, there are Secret Server web servers installed at a site in Australia, the U.S., and the U.K. The U.K. is the publisher node and Australia and U.S. nodes are the subscribers.

1. Redirect all users to the U.K. Secret Server URL.
2. Stop IIS at the Australia and U.S. sites.
3. Manually force a synchronization between the Australia subscriber and the U.K:
    a. Open SQL Server Management Studio and connect to the Australia subscriber database server.
    b. Go to **Replication > Local Subscriptions**.
    c. Right-click on one of the Secret Server subscriptions (if you used the scripts provided by Secret Server, they are called *SSPubLogs* and) and select **View Synchronization Status**.
    d. Click the **Start** button to force a synchronization.
4. Repeat sub-steps 3 and 4 for all Secret Server subscriptions.
5. Repeat sub-step 3 for the U.S. subscriber database.
6. Resolve any replication conflicts:
    a. Open SQL Server Management Studio and connect to the UK publisher database server.
    b. Go to **Replication > Local Publications**.
    c. Right-click on one of the Secret Server publications (if you used the scripts provided by Secret Server they are called SSPubLogs and SSPubEvents) and select **View Conflicts**.
    d. Examine and resolve any unresolved conflicts.
7. Generate the script to remove replication on the subscriber databases as specified above.
8. Run it to remove replication.

9. Generate the script to remove replication on the publisher database as specified above.
10. Run it to remove replication.
11. Restart IIS on the Secret Server web server in the U.K.
12. Run the Secret Server web upgrade wizard.
13. Copy the website application directory to the web servers in Australia and the U.S.
14. Use the scripts generated from the Secret Server UI to recreate replication on the publisher.
15. Push a snapshot to the Australia and U.S. subscriber databases.
16. Recreate replication at the subscribers.
17. Restart the jobs on the subscriber databases as described above.
18. Start IIS on the Australia and U.S. web servers.
19. Change redirection rules for Australia and U.S. users so they access the local WithWeb server as normal.

Other Information about SQL Server Replication

- Upgrading Secret Server with SQL Replication
- SQL Server Replication (MS Books Online)
- Snapshot Replication

Regional Availability

Below are the geo-locations and regions where Secret Server is currently deployed:

Geography	Primary Region	Failover Region
Southeast Asia	Southeast Asia (Singapore)	East Asia (Hong Kong)
Europe	Germany West Central (Frankfurt)	West Europe (Netherlands)
Canada	Canada Central (Toronto)	Canada East (Quebec City)
Australia	Australia Central (Canberra)	Australia East (New South Wales)
United Arab Emirates	UAE North (Dubai)	UAE Central (Abu Dhabi)
United Kingdom	UK South (London)	UK West (Cardiff)
United States	East US (Virginia)	West US (California)

Diagnostics

The Diagnostics section provides all the necessary information on your current Secret Server version, Operating System Configuration, Database and Secret ServerEnvironments, your current application settings, running and

scheduled jobs, allows viewing and exporting logs, clearing cache, testing events log, and exporting diagnostics.

Select **Settings>Diagnostics**, or **Admin>Diagnostics** to proceed to the Diagnostics section.

- Click **Clear Cache** at the top right to request cache clearance. Cache will be cleared in few seconds.
- Click **Test Event Log** at the top right to request test of events log.
- Click **Export diagnostics** to download the diagnostics on your local drive in *txt format.

Specifications

Select the Specification tab to find all the information about your current running Secret Server application version, including:

- Operating System Configuration - Operating System, System Type, Up Time, Memory Use, Server Name, Server Time, Server Time Zone, Domain Controller, and .NET Framework Version.
- Database Environment - SQL Server Name, Database Name, SQL Server Version, SQL Server Edition, SQL Server Collation, SQL Server Time, SQL Server Connection String, Is Replication Enabled, Replication Publisher, Replication Subscriber, and Is Synchronizing.
- Internal Site Connector Configuration - Backbone Site Connector and Backbone Class.
- Secret Server Environment - Secret Server URL, Search Indexer Background Processor, Product Version, Proxy Configuration, Max Degrees of Parallelism, Maintenance Mode, and Cached language files.
- Upgrade Information - Latest Version, Last Upgrade Details, Upgrade In progress.

App settings

Select the App settings tab to find all the settings configured on your Secret Server instance with their keys and values displayed in the grid. You can download the settings list in *csv format by clicking the download button at the top right.

See Export/Import Settings for more information.

Background Processes

Select the Background Processes tab to view all the processes running on the background displayed in the grid with the related Host and Environment names, Thread name and identity, and their last activity time specified. Click on a process in the grid to expand its details to the right. You can download the processes list in *csv format by clicking the download button at the top right.

Click Recycle Background processes at the top right to request processes recycle.

Long Running Tasks

Select the Long Running Tasks tab to view all the long running tasks displayed in the grid with the related information. You can download the tasks list in *csv format by clicking the download button at the top right.

Scheduled Jobs

Select the Scheduled Jobs tab to view all the jobs scheduled on your Secret Server instance with their last and next trigger time and current trigger state. Click on a job in the grid to expand its details to the right. You can download the jobs list in *csv format by clicking the download button at the top right.

Export logs

This feature is used to gather various logs and diagnostic information from Secret Server. It is particularly useful when troubleshooting issues or when preparing to open a support case. Collecting logs helps in identifying symptoms that match known problems, resolving non-defect issues, identifying defects, and locating root causes for faster fixes. Select the Export logs tab to collect and download collected logs for your Secret Server instance. All the latest logs are listed here in the grid with their type and status.

- Click **Collect** at the top right to collect the latest log.
- Click **Download collected logs** to download the log you have just collected on your local drive for further analysis or to provide them to Secret Server Support. The log will be downloaded as a zip file with Node log and Engine log inside.
- Click **Clear** at the top right to clear the collected log if you need to collect a fresh one.

Directory Services

Directory services are integral components of network operating systems that map the names of network resources to their network addresses. They provide a shared information infrastructure to locate, manage, and organize network resources, including volumes, folders, files, users, groups, devices, and more. Active Directory is Secret Server's native directory service.

Active Directory

Secret Serverintegrates with Active Directory (AD) to streamline user authentication and management. Key features include user authentication, user and group synchronization, Role-Based Access Control (RBAC), and Azure Active Directory Integration.

View Active Directory section for details.

Azure Active Directory

Azure Active Directory (Azure AD) is a cloud-based identity and access management solution that provides a robust and scalable way to manage user identities, authentication, and authorization. Azure Active Directory integration with Secret Server facilitates SAML-based single sign-on, streamlining user authentication and access control. It requires .NET Framework version 4.8 or later and supports advanced certificate signing settings.

View Azure Active Directory for details.

LDAP

Lightweight Directory Access Protocol (LDAP) is a widely-used protocol for accessing and managing distributed directory information services over IP networks. LDAP enables organizations to store and manage user identities, groups, and other data in a centralized repository, allowing for efficient authentication, authorization, and directory management. Secure LDAP (LDAPS) extends the protocol to provide encryption and authentication using TLS or SSL.

View LDAP for details.

AD and Secret Server Overview

> Before synchronizing or creating users, you need to create a secret to be used as the "sync secret." This secret should contain Domain Admin credentials (or an account with appropriate permissions to search and view the attributes to all your organization's users and groups).

Secret Server integrates with Active Directory (AD) to streamline user authentication and management. Here are some key points about this integration:

- User Authentication: Secret Server allows users to log in using their Active Directory credentials. This simplifies the login process and enhances security by leveraging existing AD authentication mechanisms.
- User and Group Synchronization: Before synchronizing or creating users, a "sync secret" containing Domain Admin credentials (or an account with appropriate permissions) must be created. This enables Secret Server to search and view attributes for all users and groups within the organization.
- Role-Based Access Control (RBAC): Secret Server uses RBAC to control user rights and privileges. This feature allows administrators to enforce least privilege and segregation of duties on privileged accounts. Users and groups are assigned to roles that define their permissions within the system, ensuring that access is granted strategically and securely.
- Azure Active Directory Integration: Secret Server can also integrate with Azure Active Directory, requiring .NET Framework version 4.8 or later. This integration allows for seamless management of users and groups in cloud environments.
- User Lockout Protection - When enabled, synchronization will not disable the domain's users if it would result in every user in this domain being disabled.

Active Directory Automatic User Management

Overview

When Active Directory (AD) Sync is run with the "User status mirrors Active Directory (Automatic)" option, it creates groups and users in Secret Server to mirror the organization's configured AD groups and users. A Secret Server user is created or enabled for every enabled AD user in the selected groups.

Thus, every enabled AD user in every synched group consumes a Secret Server license, whether or not they use Secret Server. As a result, an organization can end up paying for far more Secret Server licenses than they need.

AD Automatic User Management addresses this issue by automatically disabling the accounts of users who have not logged in to Secret Server in a specified number of months. This saves unnecessary licensing costs as inactive users do not count against the number of user licenses required by Secret Server.

You can configure the setting on the Edit Active Directory Configuration page. See "Configuring Active Directory" on page 503 There is a checkbox to enable or disable the feature and a textbox to set the number of months before a user is auto-disabled. The default is three, but you can set it from one to 12.

Newly-added users remain enabled until the first synchronization after the configured number of months have passed. When a user whose account has been disabled by this feature attempts to log in they automatically have their account enabled, provided there are licenses available.

Examples

Example One

1. Maria joined the company today.
2. The next AD synchronization creates a Secret Server account for Maria.
3. Maria never logs in to Secret Server because she does not need it for her job.
4. Once the defined number of months have passed, the next AD synchronization disables Maria's Secret Server account.
5. The Secret Server license used by Maria's account becomes available for use.

Example Two

This only pertains to users who have never logged into Secret Server and their account was disabled (never enabled). It does not apply to previously enabled users who have been disabled due to inactivity.

1. Joe gets added to Secret Server but never logs in.
2. The defined number of months later, Automatic User Management disables his account, freeing his license.
3. Joe gets promoted to a job that requires Secret Server.
4. Joe logs into Secret Server.
5. His account is automatically re-enabled, and he now takes up a license.
6. Joe gets demoted to his old job, which does not require Secret Server.
7. A defined number of months later, Automatic User Management disables his account, and the license is freed up once again.
8. Joe has no idea any of this has happened—the automated process is hidden from him.

Example Three

1. Rupert logs in to Secret Server several times per month.
2. The defined number of months for Automatic User Management to disable his account is never reached.

3. Rupert's account stays current and his license remains his. The entire process is invisible to Rupert.

Active Directory Credential Caching

Overview

Active Directory credential caching enables users to access Secret Server even when the domain controller is unavailable. When caching is enabled, Active Directory credentials are cached for 30 days in the on-premise editions, and for 90 days in Secret Server Cloud.

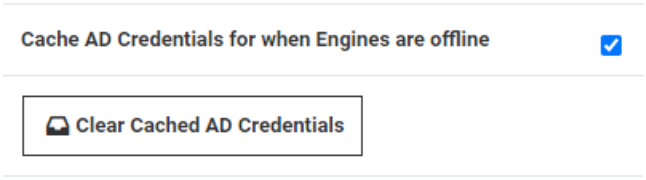
With credential caching enabled, whenever a domain user successfully logs into Secret Server, their domain password is hashed using PBKDF2, and stored in the Secret Server database along with the current time stamp.

If a domain user attempts to log in but Secret Server is unable to contact a domain controller, it falls back to the cached credentials to attempt to provide access. If the hash of the entered password matches the hash of the cached credentials and the time has not expired, the authentication will be successful.

AD Caching Configuration

AD credential caching is disabled by default, but an administrator can enabled or disable it at any time using the steps below:

1. Click **Admin** > **Configuration** and click the **Login** tab.
2. Scroll to the bottom of the window and click the **Edit** button. The tab becomes editable:



3. To enable caching, click to select the **Cache AD Credentials for when Engines are offline** check box.
4. To disable caching, click the **Clear Cached AD Credentials** button.

Auditing

Audit logs are recorded in the system log whenever cached credentials are found to be expired or when a successful login attempt has been made using cached credentials.

Active Directory Rights for Synchronization Account

Below is a listing of the Active Directory permissions required by the account used for synchronization. See "Configuring Active Directory" on page 503 for more on selecting this account.

> The locations discussed below are part of the Active Directory Administrative Center (ADAC). See Advanced AD DS Management Using Active Directory Administrative Center for information on using the ADAC.

Recommended Permissions

Object Tab

This object and all descendant objects:

- List contents
- Read all properties

Minimum Required Permissions

Note: These all require ADSI Edit - Allow (Active Directory Service Interfaces Editor) permission.

Object Tab

This object and all descendant objects:

- List contents

Properties Tab

This object and all descendant objects:

- Read objectClass

Descendant User objects:

- Read Display Name
- Read Distinguished Name
- Read E-mail-Address
- Read objectGUID
- Read Logon Name
- Read Logon Name (pre-Windows 2000)

Descendant Group objects:

- Read displayName
- Read Distinguished Name
- Read Group name (pre-Windows 2000)
- Read groupAttributes
- Read memberOf
- Read Members
- Read objectGUID

ADFS Custom Rules for Differing UPN and SAM Account Names

Overview

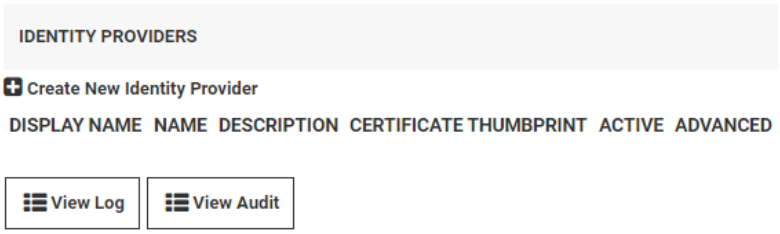
In Active Directory, when a user's sAMAccountName and UserPrincipalName (UPN) differ, you must take some steps to accommodate those differences in Secret Server. For example, suppose a user's sAMAccountName is `jsmith` and the user's UserPrincipalName is `john.smith@somedomain.com`. When Secret Server syncs with Active Directory, it obtains `jsmith` as the Secret Server login user name. However, with its standard ADFS rule passing in the UPN, Secret Server will receive `john.smith@somedomain.com` and it will not find the user.

To rectify this situation you must configure the SAML Username Attribute in Secret Server to be `customvalue`, and use three custom claim rules described below.

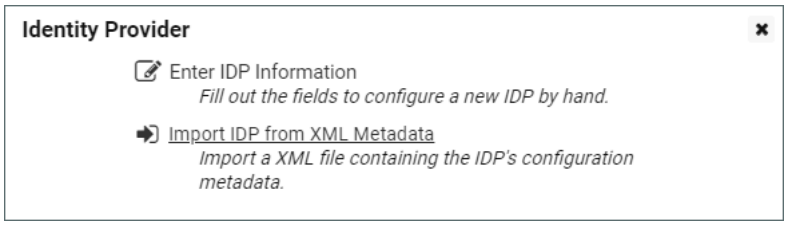
Change the SAML Username Attribute

To change the SAML Username Attribute in Secret Server, perform the following steps:

1. Click **Admin > Configuration**.
2. Click the **SAML** tab and scroll to the bottom of the window.
3. Click **Create New Identity Provider**.

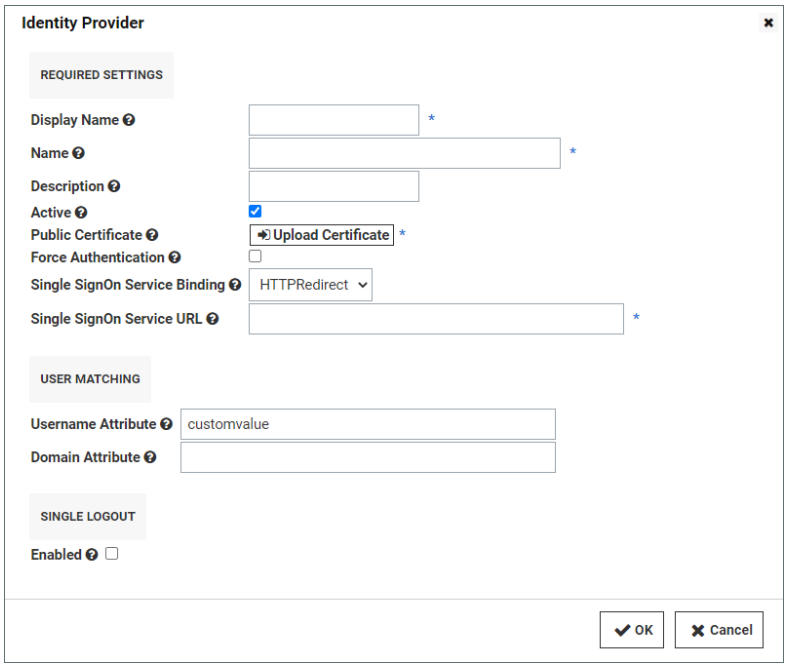


4. In the Identity Provider dialog, click **Enter IDP Information**.



5. In the next Identity Provider dialog under **User Matching**, type `customvalue` in the box next to **Username**

Attribute and click **OK**.



Create Three Rules

To create the three rules you need, open the Active Directory application and follow these steps:

1. In the **Edit Claim Rules** window, select **Add Rule**.
2. Choose **Send Claims Using a Custom Rule** as the rule template.
3. Create each rule using the information below, in the order presented.

> If you copy code directly from the webpage for pasting, please ensure that you have copied everything you need, or correct the text after pasting it.

Rule 1: Query AD for UPN and sAMaccountname Attributes

```
c:[Type == http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname,
Issuer == "AD AUTHORITY"]
 => add(store = "Active Directory", types = ("ssupn", "sswindowsaccountname"), query =
";UserPrincipalName,sAMAccountName;{0}", param = c.Value);
```

Rule 2: Obtain the Domain from the UPN

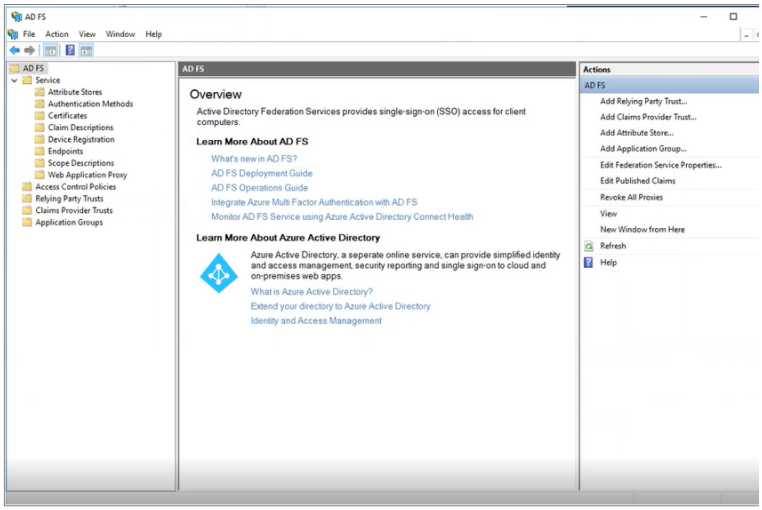
```
c:[Type == "ssupn"]
 => add(Type = "ssnewupn", Value = RegExReplace(c.Value, "^(.*?)@", ""));
```

Rule 3: Combine the sAMaccountname with the Domain

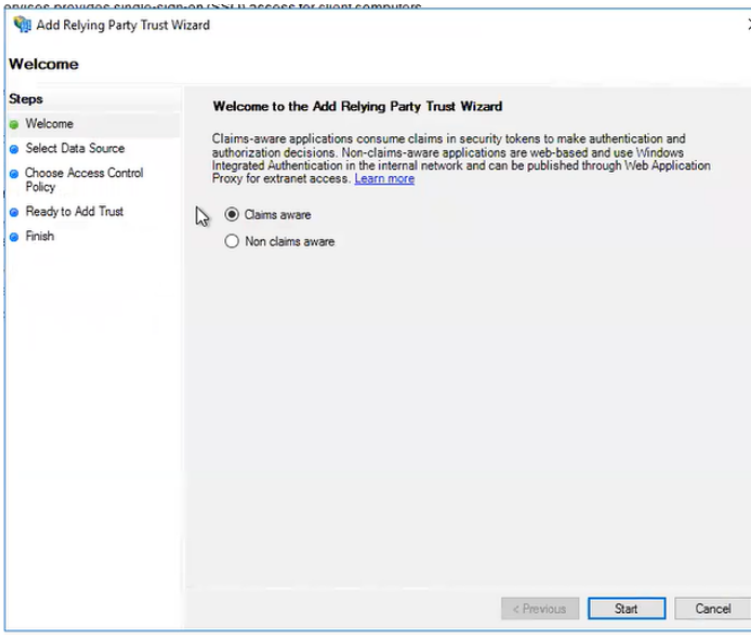
```
c1:[Type == "ssnewupn"]
 && c2:[Type == "sswindowsaccountname"]
 => issue(Type = "customvalue", Value = c2.Value + "@" + c1.Value);
```

Configuring ADFS 4.0 (Windows Server 2016)

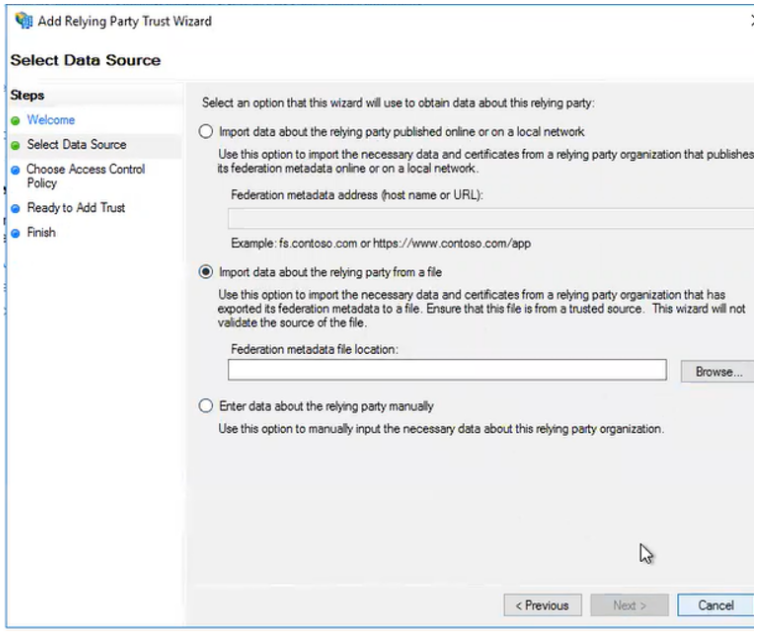
1. Go to the server on which ADFS is installed and launch the AD FS Management application.
2. Expand the Trust Relationships node and click on the Relying Party Trusts node.
3. Click on the **Add Relying Party Trust** link in the right pane to start the *Add Relying Party Trust* wizard



4. Select the **Claims aware** radio button and click the **Start** button to continue.

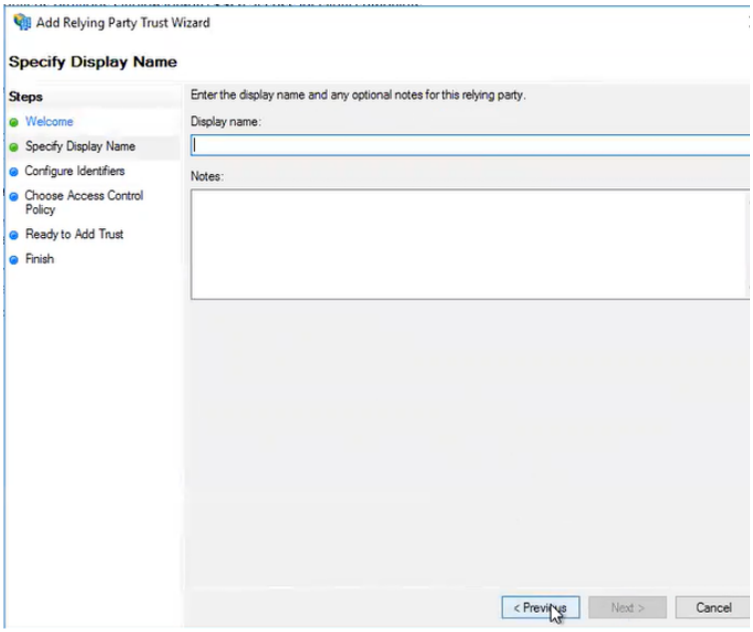


5. Select the **Import data about the relying party from a file** radio button. Browse to select the Metadata XML file you downloaded from Secret Server in earlier steps. Once uploaded, click Next to continue

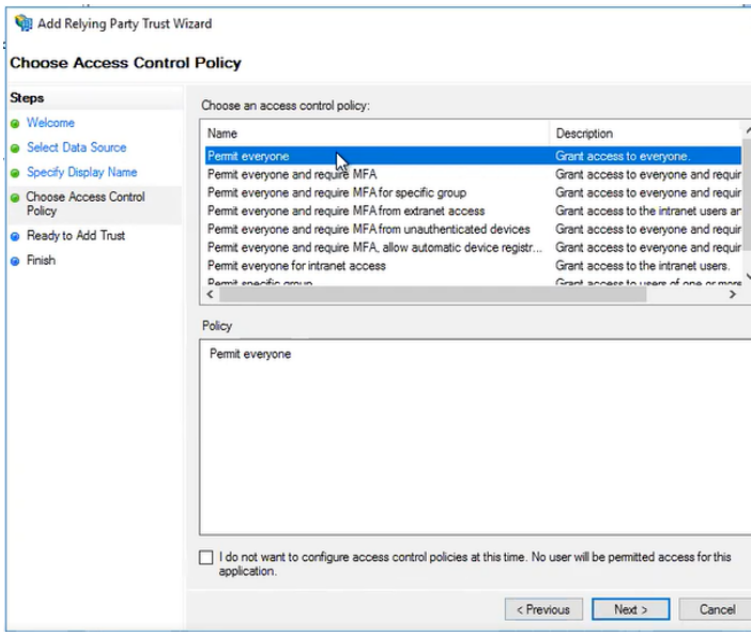


.

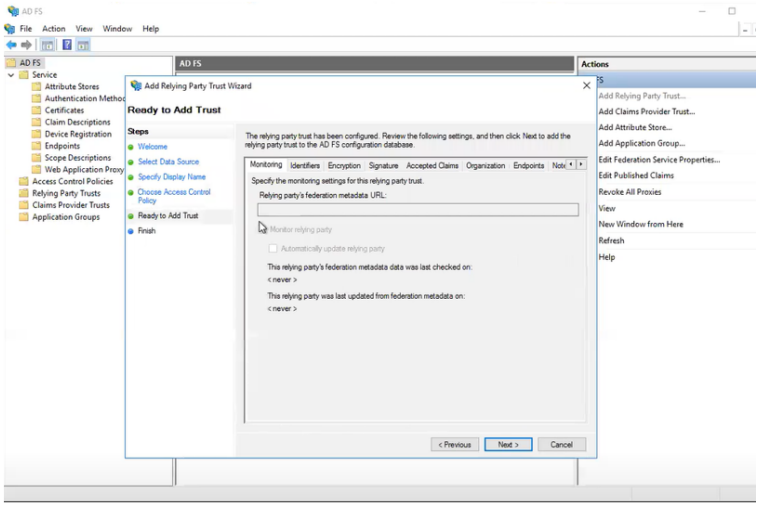
6. Choose and enter a **Display Name** for this Relying Party and any additional notes you may want. Click **Next** to continue.



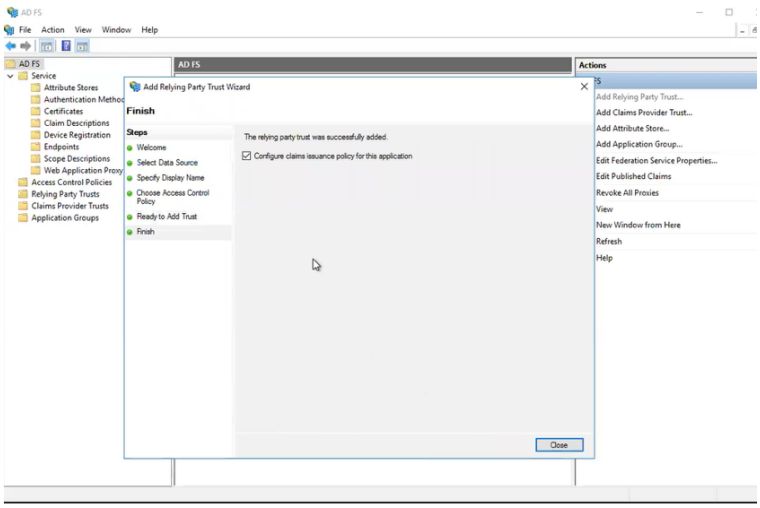
7. Choose the **Permit everyone** access control policy and then click **Next** to continue. You may optionally select another access control policy to permit only a smaller subset of users and/or require multi-factor authenticator (MFA), if needed; however, these other access control policies will not be covered in this configuration.



8. Click **Next** in the next window.



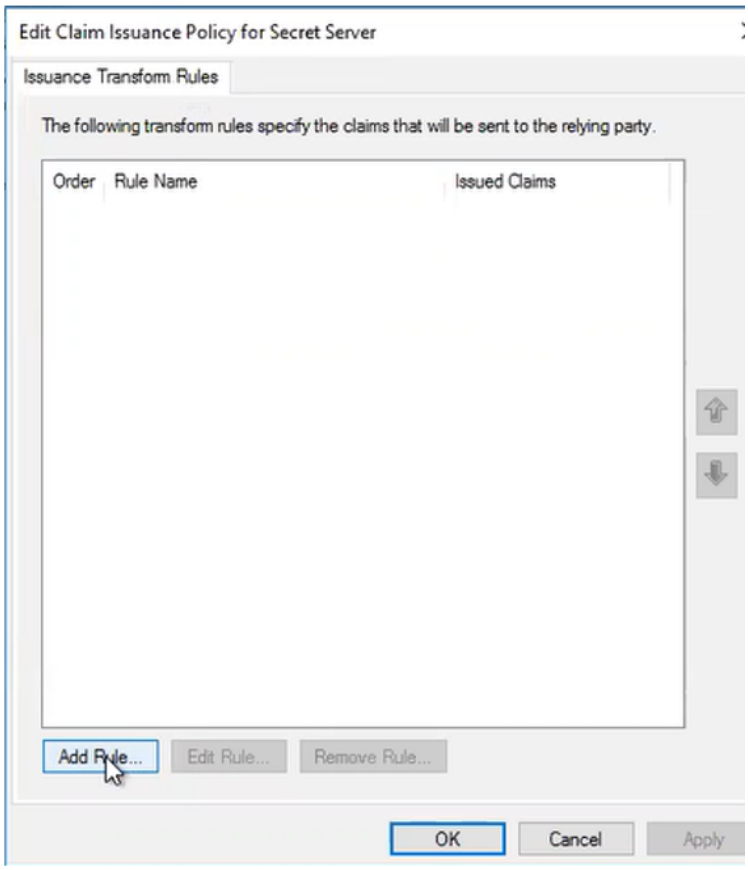
9. On the **Finish** page, make sure the **Configure claims issuance policy for this application**



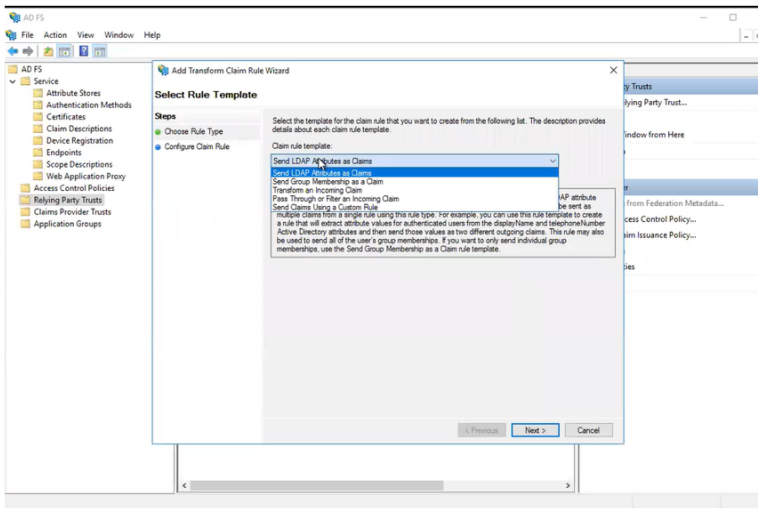
Create Claim Rules for the Secret Server Relying Party

We have now created the Relying Party trust in ADFS for Secret Server; however, we must set a claim rule so that ADFS relays information to Secret Server to describe the user's identify and authenticate.

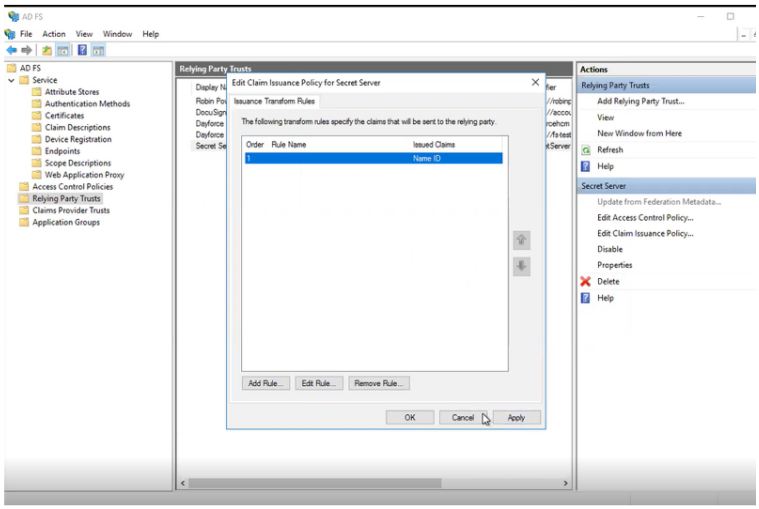
1. Let's turn our attention to the "Edit Claim Issuance Policy..." dialog window that comes up after the prior steps.



2. Click the **Add Rule** button.
3. Under **Claim rule template**, select **Send LDAP Attributes as Claims**. Click **Next** to continue.



4. Fill out the information:
    a. **Claim rule name**: Optional
    b. **Attribute Store**: select Active Directory
    c. Add an **LDAP Attribute of User-Principal-Name**
    d. Outgoing **Claim Type of Name ID**
    e. Click **Finish**
5. Click on **Apply** and **OK**.



6. Download your Metadata for the Relying Party Trust you created for Secret Server. There are several methods
    a. Navigate to **https://[YOURSERVERNAME]/FederationMetadata/2007-06/FederationMetadata.xml** to download the Metadata for your ADFS IDP. The file will automatically download.
    b. Run the PowerShell script on this page.

Configuration Parameters

Active Directory configuration can be enabled by a user with the Administer Active Directory role. To change these settings, select **Directory Services** from the **Administration** menu and then click the **Confederation** tab. Then click **Edit**.

The configuration screen offers several options:

- **Enable Active Directory Integration:** Enable or disable the Active Directory Integration feature.
- **Enable Integrated Windows Authentication:** Enable or disable the Windows integrated authentication feature.
- **Enable Synchronization of Active Directory:** Enable or disable the automatic synchronization of the selected Synchronization Groups from Active Directory. If you have manually added users and will not use the Synchronization group, do not enable this setting or manual users can be locked out.

- **Synchronization Interval for Active Directory:** Set the interval that Secret Server synchronizes its users and groups with the Active Directory.
- User Account Options:
  - **Users are enabled by default (Manual):** Secret Server users are automatically be enabled when they are synced as new users from Active Directory. If they were disabled explicitly in Secret Server, they are not be automatically re-enabled. If creating a new user causes the user count to exceed your license limit, the user is created as disabled. Secret Server
  - **Users are disabled by default (Manual):** Secret Server users are automatically disabled when they are pulled in as new users from Active Directory. If they were enabled explicitly in SS, they are not automatically re-disabled.
  - **User status mirrors Active Directory (Automatic):** When a new user is pulled in from Active Directory, they are automatically enabled if active on the domain. The exception is when this causes you to exceed your license count. For existing users, they are automatically be disabled if they are removed from all synchronization groups, deleted in AD, or disabled in AD. They are automatically re-enabled when they are part of a synchronization group and are active in AD.

Configuring Active Directory

To allow users to log in with their Active Directory (AD) credentials, you can configure your AD domain settings in Secret Server and then add users either individually or by group.

Step 1: Enabling Active Directory Integration

1. Select **Admin > Directory Services**. Alternatively, click **Access** to the left and select **Directory Services**. The Directory Services page appears.
2. Click the **Configuration** tab.
3. If **Enable Directory Services** says **No**:
   a. Click the **Edit** link next to **Directory Services**.
   b. Click the **Enable Directory Services** check box.
   c. Click the **Save** button.

Step 2: Adding a Domain

1. Select **Admin > Directory Services**. The Domains tab of the Directory Services page appears.
2. Click the **Add Domain** button and select **Active Directory Domain**. The Active Directory popup appears.
3. Type the FQDN and friendly name in their text boxes.
4. If you wish to use Secure LDAP, enable the **Use LDAPS** checkbox.
5. If you have an existing sync secret, click the **No Secret Selected** link next to **Synchronization Secret** to select the secret containing the username and password for connection. Otherwise, click the **Create New Secret** link to create one.
6. Click the **Site** dropdown list to select the site for the AD location.

7. Click the **Multifactor Authentication** dropdown list to select the desired MFA, if any.
8. Click the **Validate & Save** button.

Now you are ready to add individual users or groups of users for access to Secret Server with AD credentials. See the relevant section below for instructions.

Step 3: Setting Up Synchronization Groups

Once you add a domain, add the applicable synchronization groups users from these groups are available for login, subject to synchronization settings. These groups can also be used to assign permissions, roles, and sharing. To add groups:

> If the specific group does not exist, one can be created by your Active Directory administrator. If you create domain users manually or want to convert local users to domain users, see the corresponding sections below before setting the synchronization group.

1. Click the **Domain Name** link on the **Domains** tab. The page for that domain appears.
2. Click the **Groups** tab.
3. Click the **Edit** link next to **Synchronized Groups**. The top Synchronized Group table and search box represent the available sync groups. The Select Groups list shows all the groups belonging to the domain (those with selected check boxes).
4. Type the name of the desired group in the **Search domain for groups** search text box or scroll down the list to find a group.
5. Click the group. The users belonging to the group appear below in a list to provide a preview to aid selecting groups.

Step 4: Adding or Removing Groups

1. Click the check box in the **Select Groups** list to add or remove groups from the domain.
2. If you want to remove an entire group instead of not selecting it (leaving it for later use), click the **Remove** link next to the group name in the **Synchronized Groups** table.
3. Click the **Save** button.

> Enabled users count towards your Secret Server user licensing.

Step 5: Enabling Active Directory Synchronization

> Two safeguards are built into the synchronization process to prevent the system from erroneously disabling users as a result of receiving incomplete information from the directory server. If communication with a directory server returns an error, then no membership changes are made to users from that directory. Similarly, if communication with a directory server returns zero users, no membership changes are made to users from that directory.

1. Return to the **Directory Services** page.
2. Click the **Configuration** tab.
3. Click to select the **Enable User Synchronization** check box. Additional settings appear.
4. Choose how often you want Secret Server to sync with AD by configuring the **Synchronization Interval**. The default value is one day.
5. Click the **User Account Options** Dropdown list to select a default status for users. See below for a description of each option. We recommend selecting **Users are disabled by default (Manual)** for initial testing. The options are:
   - **Users are enabled by default (Manual)**: Secret Server users are automatically enabled when they are synced as new users from AD. If they were disabled explicitly in Secret Server, they are not automatically re-enabled. If creating a new user will cause the user count to exceed your license limit, the user created disabled. Secret Server
   - **Users are disabled by default (Manual)**: Secret Server users are automatically disabled when they are pulled in as new users from AD. If they were enabled explicitly in SS, they are not automatically re-disabled.
   - **User status mirrors Active Directory (Automatic)**: When new users are pulled in from AD, they are automatically enabled if active on the domain. The exception is when this will cause you to exceed your license count. For existing users, they are automatically disabled if they are removed from all synchronization groups, deleted in AD, or disabled in AD. They are automatically re-enabled when they are part of a synchronization group and are active in AD. See "Active Directory Automatic User Management" on page 491.
6. Change the **Days to Keep Operational Logs** text box to set the period to keep AD-related logs that might contain PII. Secret Server automatically deletes logs older than that (in days).
7. Click the **Save** button.

Step 6: Running Active Directory Synchronization

From the **Directory Services** page, click the **Sync Now** button to run a sync. As the sync progresses, you can click the **Refresh** button to monitor the logs until you see the message **Completed Domain synchronization for all domains**.

Converting Local Users to Domain Users

Local users can be converted to a domain user in a one-way irreversible process. This feature helps existing customers with extensive groups and permissions setup for a local user that they want to convert to an Active Directory user. The page can be accessed on the **Administration > Users** page by clicking the **Migrate to AD** button. For the conversion to work, the domain user must not exist within Secret Server. The username is changed to match the domain user throughout the system.

Creating Active Directory Users

Active Directory users can be created manually by a user that has the Administer Users role. You can do this by going to **Administration > Users**, then clicking the **Create New** button. See "Creating Users" on page 1273.

Enabling and Disabling Active Directory Users

If you selected a manual setting for **User Account Options**, you can now enable or disable your AD users' access to Secret Server:

1. Go to **Admin > Users**. The Users page appears.
2. To enable users:
   a. Click to select the **Show Inactive Users** check box.
   b. Click to select the check box next to the users to enable.
   c. Click The **Bulk Operation** dropdown list and select **Enable Users**.
3. To disable users, use the same process, selecting **Disable Users** from the **Bulk Operation** dropdown list.

Setting up SAML SSO for Active Directory

How to set up Single Sign-On (SSO) for users synced between an Active Directory domain server and a Secret Server user list.

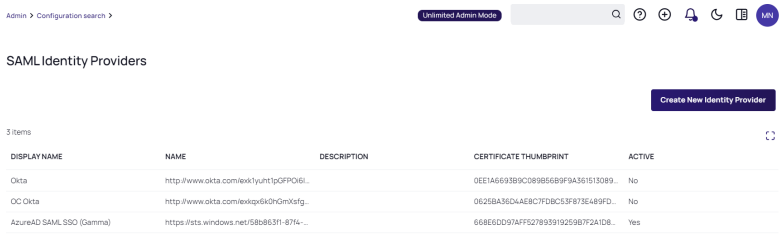
The interface and workflows for Active Directory Federation Services (ADFS) Server are subject to change. For more current workflow and interface references, please refer to the Microsoft ADFS Server documentation.

ADFS Server

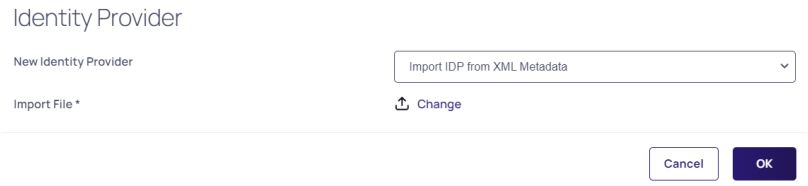
1. On your ADFS server, browse to your Secret Server instance and sign in.
2. Download the `SecretServerSAMLMetadata.xml` file from `[YourSecretServerInstance.Name]/samlmetadata`.
3. Open **Active Directory Federation Services Management**.
4. Under **Trust Relationships**, click **Add Relying Party Trust** to add your service provider information.
5. In the **Add Relying Party Trust** wizard, click **Start**.
6. Click **Import data about the relying party from a file**.
7. Browse to select the metadata XML file you downloaded earlier and click **Next**.
8. Enter a display name of your choice and click **Next**.
9. Decide whether to configure multi-factor authentication and click **Next**.

Secret Server

1. In Secret Server, click **Admin > SAML Identity Providers**, and click **Create New Identity Provider**.



2. Click **Import IDP from XML Metadata** and select the ADFS metadata you downloaded. If you don't see the file, you might need to change the metadata filetype to xml.



Adding Users to ADFS

For users to be authenticated by the SSO workflow you are setting up, Secret Server usernames must match domain AD usernames. If you manually add usernames to Secret Server or AD, you must inspect them carefully to ensure that they match. You can also use Secret Server Discovery to sync Secret Server usernames in bulk with AD usernames.

Once a username matches in both systems, the user can log into their desktop computer using their AD credentials and then browse to Secret Server without being prompted again for authentication.

> If you have accounts in which the sAMAccountName differs from the UPN name, you can create custom rules to accommodate the differences. See the Directory Services section of the Secret Server documentation.

Common Errors

If you encounter any of the errors below, check that the **RelyingPartyTrust Rule** on the ADFS server has both the message and assertion signed. By default, only the assertion is signed.

- "Attempt to login via SAML from identity provider had no signed responses or assertions"
- "Attempt to login via SAML with unsigned request"
- "Attempt to login via SAML with unsigned assertion"

If you encounter the error, "SAML Response signature message from IDP failed verification," it means that Secret Server cannot decrypt the assertion message from the IDP (ADFS) because the public certificate thumbprint is incorrect. To fix this issue, follow the steps below.

1. Download the ADFS certificate, upload it to Secret Server (**Admin > Configuration > SAML** tab) and edit the IDP configuration.
2. Check the token-decrypting in ADFS to verify the certificate.
3. Use the Get-AdfsCertificate cmdlet to retrieve the certificates listed below that ADFS uses, and check that they are appropriately identified as primary (**IsPrimary** is set to **True**):
   - A primary token-signing certificate is used to digitally sign outgoing claims.
   - A primary token-encrypting certificate is published in federation metadata for use by trusted claims providers.
   - Information card signing and service communications certificates are always primary.

Syncing and Authenticating AD Users via a Distributed Engine

Local Versus Distributed Engine Sites

Secret Server connects to the domain: from the Web server *or* routed through a distributed engine. If your Web server can reach your domain without issue, then using the local site option is recommended. When a user authenticates or AD synchronization is run, the connection to the domain is from the Web server. If your Web server cannot connect to the target domain, if it is a VM in a cloud environment for example, you can setup an engine on-premises and assign it to the domain. When a user authenticates, Secret Server routes the domain calls through the on-premises engine, eliminating the need for site to site connections or persistent VPNs. Review the Distributed Engine guide for steps on setting up sites and engines.

> The Active Directory secret is used to synchronize users and groups, it requires permission to search and view the attributes of the users and groups. If you plan on using discovery, the account also needs permissions to scan computers on the network for accounts.

To setup AD to sync from a DE:

1. Create a synced secret. Before synchronizing or creating users, create a secret for use as the sync secret. This secret should contain user-level credentials (or an account with appropriate permissions for read access to all your organization's AD objects).
2. Specify the domain to authenticate against:
   a. Before synchronizing or creating users, you must first specify which domains Secret Server can authenticate against. Secret Server can synchronize with any number of domains.
   b. Go to **Admin > Active Directory**. The Active Directory Configuration page appears.
   c. Click the **Edit Domains** button.
   d. Click the **Create New** button. The Active Directory Domain page appears.
   e. Type the domain information that you want to authenticate to.
   f. Click the **Link a Secret** selection button.
   g. Click the **Sync Secret** list to select the AD secret you created earlier.

> If you do not have a secret setup yet, click the **Create New Secret** link to create your AD secret.

> The AD sync secret is used to synchronize users and groups. It requires permission to search and view the attributes of the users and groups. If you plan on using Secret Server discovery, the account will also need permissions to scan computers on the network for accounts.

h. Click the **Save and Validate** button.

3. Set up the synchronization groups:
   a. Once the domain has been added, go to **Admin > Active Directory**. The Active Directory Configuration page appears.
   b. Click the **Edit Synchronization** button. The Synchronization Edit page appears. The Available Groups represent all accessible groups on the specified AD domain. You can preview the user membership with the Group Preview control.
4. Select the desired group from the Available Groups that contains the AD accounts for users you would like to create in Secret Server.
5. Configure AD:

> See "Configuration Parameters" on page 502 for more information.

   a. Go to **Admin > Active Directory**. The Active Directory Configuration page appears.
   b. Click on the **Edit** button. The Edit Active Directory Configuration page appears.
   c. Click to select the **Enable Active Directory Integration** check box.
   d. Click to select the **Enable Synchronization of Active Directory** check box.
   e. Click the **Save** button.
   f. Turn on AD sync.

Azure Active Directory

Azure Active Directory (Azure AD) is a cloud-based identity and access management solution that provides a robust and scalable way to manage user identities, authentication, and authorization. As a critical component of Microsoft's cloud ecosystem, Azure AD enables organizations to securely manage access to applications, services, and data, both on-premises and in the cloud. With features like single sign-on (SSO), multi-factor authentication (MFA), and conditional access, Azure AD helps protect against cyber threats and ensures that only authorized users have access to sensitive resources. Additionally, Azure AD provides advanced tools for identity governance, risk detection, and compliance, making it an essential platform for businesses seeking to strengthen their security posture and streamline user management.

Configuration Parameters

Azure Active Directory (*Azure AD*) configuration can be enabled by a user with the Administer Active Directory role. To change these settings, navigate to **Admin | Directory Services**, click the **Domain Name** associated with your Azure AD directory, and then click **Edit**.

When creating a new directory, the required configuration screen settings have the following fields:

- **Domain Name:** A friendly display name for the Azure Directory. When an existing Azure AD Domain is edited, you can still view and edit the Tenant ID, Client ID, and Client Secret fields, or synchronize secret.
- **Active:** Enable or disable the Azure Active Directory domain integration.
- **Tenant ID:** Globally unique identifier (GUID) value assigned to the Azure AD directory.
- **Client ID:** Globally unique identifier (GUID) value assigned to the Client Secret upon creation. Portal will also reference this as the *Application ID* or *App ID*.
- **Client Secret:** Unique, generated string for the Client Secret. *This value can only be retrieved upon creation.*
- **Synchronization Secret:** Select or create a secret for synchronization. If the Synchronization Secret is set, the Tenant ID, Client ID, and Client Secret will be taken from the Synchronization Secret. If the Synchronization Secret cleared, the Tenant ID, Client ID, and Client Secret fields can be edited again, but once an Azure AD domain is saved with a Synchronization Secret set, the Tenant ID, Client ID, and Client Secret will not be editable anymore.

Optionally you can also configure the following:

- **Multifactor Authentication:** Drop-down selection for the desired MFA.

Create Azure App Registration

The steps provided can be used to create the app registration required for configuring Azure Active Directory integration.

This integration requires .NET Framework version 4.8 or later.

For this procedure, you need a local account with the Administer Active Directory role.

Azure Portal Method

Create the Application Registration

Follow the steps in Register an application with the Microsoft identity platform to register an app on Azure Portal.

Use Delinea Secret Server as a **Name** for your application, and `https://<Your Secret Server URL>/signin-oidc` as **Redirect URL**.

Once the app registration is created, take note of the Application (client) ID and Directory (tenant) ID, these will be needed for Secret Server configuration.

Add Client Secret to the Application Registration

Follow the steps in the Add a client secret section of the Register an application with the Microsoft identity platform guide, to learn how to add a **Client Secret** to the application registration.

Use `Secret Server` as the **Description** and record the text string in the **Value** column for that secret when it is successfully added.

Add API Permissions to the Application Registration

Follow the steps in Configure an application to expose a web API guide to add API Permissions to the Application Registration.

> This requires a local account with at least one of these roles: **Administer Active Directory**, **Unlimited Administrator**, or **Administer Configuration Unlimited Admin**.

1. Click **API Permissions** in the left panel in the **Manage** section. The **API Permissions** page appears.
2. If any default permissions appear in the unlabeled configured permissions table, click the **...** button and select **Remove Permission**.
3. Select the **Add a Permission** button. The **Request API Permissions** page appears.
4. Select the **Microsoft Graph** panel button. A wizard begins.
5. Choose **Application Permissions** when asked **What type of permissions does your application require?**. The **Select Permissions** section appears.
6. In the search text box, type Group. A **GroupMember** section appears.
7. Click to expand the section.
8. Check the box for the following application permissions:
   - **Group.Read.All**
   - **GroupMember.Read.All**
   - **Member.Read.Hidden**
   - **User.Read.All**
9. Choose **Delegated Permissions** when asked "**What type of permissions does your application require?**". The **Select Permissions** section appears.
10. Check the box for the following Delegated permissions:
    - **Group.Read.All**
11. Select the **Add Permissions** button. A prompt appears.
12. Click **Yes** to grant consent to all accounts in the directory. You will receive a notification for grant consent, and a green check mark appears in the Status column on the **Configure Permissions** page.

Script Method

The script below is provided as-is, and future use may require adjustment if Microsoft changes the AzureAD PowerShell module.

> At the time of writing, there is no command in the AzureAD module granting admin consent to the app. That step has to be performed via the Azure Portal.

```
<#
```

```
    Connect to your tenant
#>
$tenantId = ''
Connect-AzureAd -TenantId $tenantId
<#
    Variables - Adjust for your environment/requirements
#>
$appName = "Delinea Secret Server"
$appRedirect = "https://vault.company.com/signin-oidc"
<#
    DO NOT CHANGE
#>
$appPerms = 'Group.Read.All','GroupMember.Read.All','Member.Read.Hidden','User.Read.All'
<#
    Pull the Service App ID for Microsoft Graph
#>
$msGraphService = Get-AzureADServicePrincipal -Filter "DisplayName eq 'Microsoft Graph'"
<#
    Create object for Resource Access - assigning app role permissions
#>
$msGraphResourceAccess = New-Object -TypeName
"Microsoft.Open.MSGraph.Model.RequiredResourceAccess"
$msGraphResourceAccess.ResourceAppId = $msGraphService.AppId
<#
    This grabs the ID for each permission listed in $appPerms variable
#>
$permissions = $msGraphService.AppRoles.Where({$_.Value -in $appPerms})
foreach ($p in $permissions) {
    $appPermissions = New-Object -TypeName "Microsoft.Open.MSGraph.Model.ResourceAccess" -
ArgumentList $p.Id,"Role"
    <# Add the role to the resource access object #>
    $msGraphResourceAccess.ResourceAccess += $appPermissions
}
<#
    Create the App Registration
#>
$paramsApp = @{
    DisplayName = $appName
    Web = @{ RedirectUris = $appRedirect }
    RequiredResourceAccess = $msGraphResourceAccess
}
$thycoticApp = New-AzureADMSApplication @paramsApp
<#
    Create the Client Secret and assign to the App Registration created
    !!NOTE!! MSGraph only supports the expiration being set to 2 years, no configuration
option is provided
#>
$paramsClientSecret = @{
    ObjectId = $thycoticApp.Id
    PasswordCredential = @{ displayName = "#{PRODUCTNAME}# $(Get-Date -Format yyyy-MM-
dd)"}
}
$clientSecret = New-AzureADMSApplicationPassword @paramsClientSecret
```

```
<#
    Output object data needed for configuring$1#{PRODUCTNAME}#$2
#>
[pscustomobject]@{
    Details = "These values required for #{PRODUCTNAME}# Configuration"
    TenantId = (Get-AzureADTenantDetail).ObjectId
    ClientID = $thycoticApp.AppId
    ClientSecret = $clientSecret.SecretText
} | Format-List
```

Configure Azure Active Directory Domain

The steps below are used for adding an Azure Active Directory configuration to Directory Services.

> For this procedure, you need a local account with the Administer Active Directory role.

Add Azure Active Directory Domain

1. Navigate to **Admin > Directory Services**.
2. Click the **Add Domain** button.
3. Click the **Azure Active Directory Domain**.
4. Set the following fields:
   - **Domain Name:**A friendly display name for the Azure Directory.
   - **Active:** Enable or disable the Azure Active Directory domain integration.
   - **Tenant ID:** The tenant GUID for the Azure Active Directory domain.
   - **Client ID:** The client GUID for the registered application for the Azure Active Directory domain.
   - **Client Secret:** The client secret for the registered application for the Azure Active Directory domain.
   - **Multifactor Authentication:** (Optional) Click on the dropdown list to select your desired MFA.
5. Click the **Validate & Save** button. Once validation completes, you will see the Friendly domain name listed.
6. Click the name of the new domain to open the configuration page.
7. Click the **Groups** tab.
8. Click the **Edit** link next to **Synchronized Groups**.
9. Scroll to or search for each desired group containing users you want to sync in the **Select Groups** table. Ensure each group's check box is checked.
10. Click the **Save** button to save your changes. You will now see the selected groups in the Synchronized Groups table.
11. Click the **Directory Services** breadcrumb link at the top of the page to navigate back to the Directory Services page.
12. Click the **Sync Now** button to sync the directory groups.

Setting up Entra ID for SAML

For the detailed information on how to setup SAML-based single sign-on for Secret Server in Entra ID, see Microsoft's Enable single sign-on for an enterprise application.

> Users must have Entra Domain Services already configured to use Entra ID.

Adding Users to Single Sign-On in Entra ID

Follow the steps in Register the user account guide to learn how to register a user account for your application.

> If you have accounts in which the sAMAccountName differs from the UPN name, you can create custom rules to accommodate the differences. See "Directory Services" on page 490.

Entra ID Configuration Steps

> For more information on how to setup SAML-based single sign-on for Secret Server in Entra ID, see Microsoft's Enable single sign-on for an enterprise application .

> You must have SAML already setup in Secret Server with a valid certificate. See the Setting up Secret Server section in "Configuring SAML Single Sign-on" on page 422.

Follow the steps in Configure SAML setting to register a user account for your application.

Advanced Settings

The following Secret Server Identity Provider Advanced Settings can be configured in Entra ID:

- Advanced certificate signing options in a SAML token
- SAML Request Signature Verification
- Configure SAML token encryption

> If you apply advanced certificate signing settings to the Secret Server IdP application in Entra ID, return to the Identity Providers page in Secret Server and click the **...** button next the provider and select **Advanced Settings** to apply the same settings.

> Custom claims can be configured within the Azure Enterprise Application in order to match the incoming claim to the Secret Server username.

Entra ID Configuration Steps

> For more information on how to setup SAML-based single sign-on for Secret Server in Azure Active Directory, see Microsoft's Enable single sign-on for an enterprise application .

> You must have SAML already setup in Secret Server with a valid certificate. See the Setting up Secret Server section in "Configuring SAML Single Sign-on" on page 422.

1. Log into your portal.azure.com account.
2. Navigate to **Entra ID**.
3. Navigate to **Enterprise Applications**.
4. Select **New Application**.
5. Select **Non-gallery application**.
6. Give your new IdP application a name and click **Add**.
7. Click **Single sign-on**.
8. In the dropdown, select **SAML-based Sign-on**.
9. If you haven't done so already, download the Secret Server metadata file named `SecretServerSAMLMetadata.xml` from `[YourSecretServerInstance.Name]/samlmetadata`:
   a. In Secret Server, navigate to **Administration**.
   b. Type SAML in the combination box. The Section Matches section populates with sections matching what you typed.
   c. Click the **SAML Service Provider Settings** link. The SAML Service Provider Settings page appears.
   d. Click the **Download Service Provider Metadata (XML)** button. The SecretServerSAMLMetatdata.xml file downloads to your browser's default location.

   > For more information on setting SAML up in Secret Server, please see See the Setting up Secret Server section in "Configuring SAML Single Sign-on" on page 422.
10. Click **Upload metadata file** and upload the Secret Server Metadata file you previously downloaded.
11. Click **Save**.
12. Scroll down and click **Metadata XML** to download the metadata for this application.
13. Go back to **Entra ID** and click on **App registrations**.
14. Select your Azure Identity Provider (IdP) application.

    > If you don't see the application immediately, you might need to click View all Applications.
15. Click **Settings > Properties**, then enter the Logout URL field for your instance. The form for this URL will be: https://`[YourSecretServerInstanceName]/saml/SLOService.aspx`.
16. Click **Save**.
17. Return to the **Configuration** search box and type `Identity Providers`.
18. Click on the **Identity Providers** link that appears in the **Section Matches** section. The SAML Identity Providers page appears.

19. Click the **Create New Identity Provider** button. The Identity Provider popup appears.
20. Click the **New Identity Provider** dropdown list and select **Import IDP from XML Metadata**. An Import File control appears.
21. Click the upload icon and select the XML file you downloaded earlier. If you do not see the file where it should be, ensure the file type is set to XML.
22. Click the **OK** button.

Adding Users to Single Sign-On in Azure AD

For users to be authenticated by the SSO workflow you are setting up, Secret Server usernames must match Entra ID usernames. If you manually add usernames to Secret Server or Entra ID, you must inspect them carefully to ensure that they match. You can also use Secret Server Discovery to sync Secret Server usernames in bulk with Entra ID usernames.

1. Log into your portal.azure.com account.

   Navigate to **Entra ID > Enterprise Applications** and select your IdP from the list
2. Select **Users** and groups and **Add User**.
3. Click **Users and groups/None Selected**.
4. Search for the user you want to add to your SAML workflow. (Note that any users added must also exist in your Secret Server instance. Usernames must match between the systems).
5. Click **Select** at the bottom, then **Assign**.

Once a username matches in both systems, the user should be able to use the Single Sign-On workflow. To test this, log into Entra ID as the user, then browse to your Secret Server instance. The user should be logged into Secret Server automatically without being prompted again for login credentials.

> If you have accounts in which the sAMAccountName differs from the UPN name, you can create custom rules to accommodate the differences. See "Directory Services" on page 490.

Advanced Settings

The following Secret Server Identity Provider Advanced Settings can be configured in Entra ID:

- Require Signed SAML Response
- Require Signed Assertion
- Require Signed Assertion Or Signed SAML Response

Below are the steps to configure the settings in Entra ID:

1. Log in to portal.azure.com.
2. Navigate to **Entra ID > Enterprise Applications**.
3. Select your IdP, then click **Single sign-on**.
4. Scroll down and check the box for **Show advanced certificate signing settings** checkbox.

5. Click the drop-down arrows to reveal options. These advanced options correspond with advanced options in Secret Server.
6. Click **Advanced Settings** next to your identity provider.
   - Require Signed SAML Response
   - Require Signed Assertion
   - Require Signed Assertion Or Signed SAML Response

> If you apply advanced certificate signing settings to the Secret Server IdP application in Entra ID, return to the Identity Providers page in Secret Server and click the ... button next the provider and select Advanced Settings to apply the same settings.

LDAP

Lightweight Directory Access Protocol (LDAP) is a widely-used, open-standard protocol for accessing and managing distributed directory information services over IP networks. LDAP enables organizations to store and manage user identities, groups, and other data in a centralized repository, allowing for efficient authentication, authorization, and directory management. Secure LDAP (LDAPS) extends the protocol to provide encryption and authentication using Transport Layer Security (TLS) or Secure Sockets Layer (SSL), ensuring confidentiality and integrity of data in transit. OpenLDAP is a popular, open-source implementation of the LDAP protocol, providing a scalable and customizable directory server solution that supports a range of platforms and applications. With its flexibility, scalability, and security features, LDAP remains a fundamental technology for identity management, authentication, and authorization in modern computing environments.

Syncing with OpenLDAP Directory Service

Introduction

OpenLDAP is a free, open source version of the Lightweight Directory Access Protocol (LDAP) developed by the OpenLDAP Project. This topic describes syncing OpenLDAP to Secret Server.

> This feature is only supported by the new interface. The classic interface does not support OpenLDAP Directory Services.

Unsupported and Difficult Use Cases

Anonymous User Authentication

We do not support anonymous user authentication:

When creating an OpenLDAP directory service, "Anonymous" is a supported authentication method. When this is chosen, Secret Server connects anonymously to the OpenLDAP directory service as configured during the synchronization process and creates any users found on the directory service.

When anonymous is selected, a secondary authentication option, "User Authentication," appears, which is the method used when the synchronized users attempt to authenticate to Secret Server. In short, user authentication cannot be anonymous because Secret Server does not allow anonymous access.

The valid options for user authentication when anonymous is selected for the synchronization process are "Basic," "Kerberos," or "No Authentication." "No Authentication" supports using an OpenLDAP directory service as a user directory while enabling alternative methods of authentication, such as SAML.

Duplicate User Attributes

We do not support configurations where using different attributes yield users with the same username, GUID, or user principal name (email address format–not necessarily an actual email address). These must all be unique to each user. If a duplicate exists, it may result in odd, unpredictable behavior from the application.

OpenLDAP Password Changer Servers Using a DNS ANAME Record

Overview

Any OpenLDAP server that uses a DNS ANAME alias record in its Transport Layer Security (TLS) certificate requires an additional registry entry for the Microsoft ADSI library to successfully do a TLS handshake. For example, this problem directly impacts connections with Okta LDAP servers.

Registry Entry

The registry entry is:

```
(DWORD) HKLM\SYSTEM\CurrentControlSet\Services\ldap\UseHostnameAsAlias = 1
```

Example

We want to integrate with dev-99352743.ldap.okta.com (or any variation). We run Nslookup:

> Nslookup is a command-line tool used for querying the Domain Name System (DNS) to obtain the mapping between a domain name and its IP address or other DNS records. It is commonly used to retrieve detailed information about a specified domain, which is essential for troubleshooting DNS-related problems.

```
C:\Program Files\SafeNet\LunaClient>nslookup
dev-99352743.ldap.okta.com
Server: dns-cac-lb-02.rr.com
Address: 2001:1998:f00:2::1

Non-authoritative answer:
Name:   ok12-ldapi-6062af7f5304741c.elb.us-west-2.amazonaws.com
Addresses: 44.234.52.16
44.234.52.15
44.234.52.17
Aliases: dev-99352743.ldap.okta.com
ok12.ldap.okta.com
```

From this we glean:

ANAME record: dev-99352743.ldap.okta.com

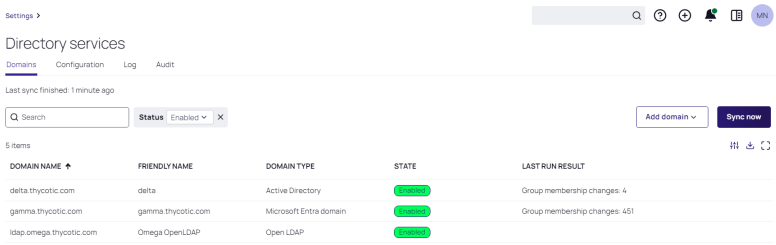
CNAME record: ok12-ldapi-6062af7f5304741c.elb.us-west-2.amazonaws.com

> A "non-authoritative answer" means that the answer is not fetched from the authoritative DNS server for the queried domain name. Instead, it is obtained from a DNS server that has the information in its cache or has obtained it from an authoritative server and is providing the information as a best guess.

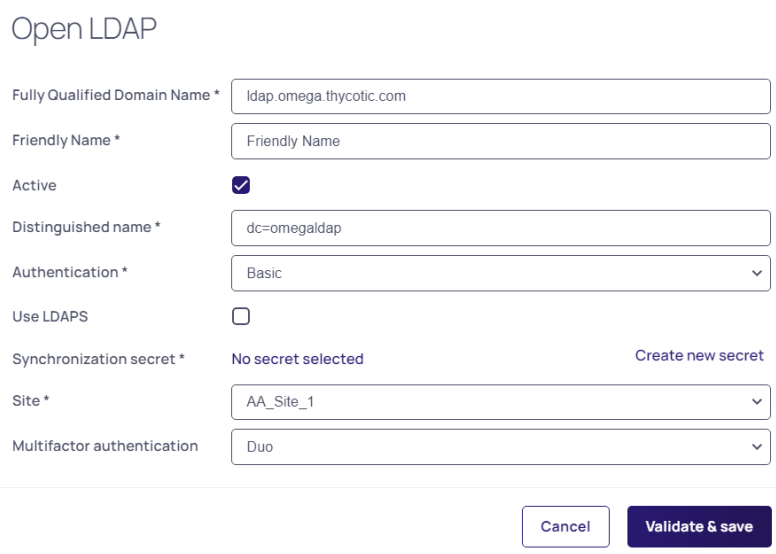
Thus, without the registry entry, the Microsoft library connects to the CNAME, and the TLS handshake fails. With the registry entry, the Microsoft library connects to the ANAME and the TLS handshake succeeds.

Procedure

1. Create a secret in Secret Server of type **OpenLDAP Account**. This sync secret is used to synchronize users and groups. It requires permission to search and view the attributes of the users and groups. If you plan on using Secret Server discovery, the account will also need permissions to scan computers on the network for accounts. Complete these parameters:
   - Domain. Example: `ldap.omega.thycotic.com`
   - Username. Example: `cn=ldap,dc=omegaldap,dc=local`
   - Password
2. Go to **Admin > Directory Services**. The Directory Services page appears:



3. Click the **Add Domain** dropdown list and select **OpenLDAP Domain**. The OpenLDAP popup appears:



4. Type the domain's FQDN in the **Fully Qualified Domain Name** text box. For example: `ldap.omega.thycotic.com`.
5. Type any name you desire in the **Friendly Name** text box.
6. Ensure the **Active** check box is selected.
7. Type the distinguished name (node path) in the **Distinguished Name** text box. For example: `dc=omegaldap,dc=local`
8. Click the **Authentication** dropdown list to select either the **Basic** or **Anonymous** authentication method.
   - Basic authentication requires that valid credentials are assigned as the sync secret. Those credentials are used to authenticate to the OpenLDAP system on each sync.
   - Anonymous authentication does not require valid credentials and removes the Synchronization Secret section. Instead, it exposes a User Authentication field.

   The Kerberos authentication method probably works but has not been tested by Delinea.
9. Basic authentication:
   a. Click the **No Secret Selected** link in the Synchronization Secret section. The Select Secret popup appears.
   b. Navigate to and select the secret you created earlier. The moment you click it, the popup disappears and the secret name appears in the Synchronization Secret section.
10. Anonymous authentication: Click the **User Authentication** list to select **Basic** or **No Authentication**. This sets which authentication method to use when users who are synced anonymously try to authenticate:
    - Basic authentication requires valid OpenLDAP account credentials.
    - No authentication is for when customers want users synced from OpenLDAP but use authentication through another service, such as SAML. We do *not* support anonymous authentication for security reasons.
11. Click to select the **Use LDAPS** check box if you intend to use secure LDAP.
12. Click the **Site** dropdown list to select your site.
13. Click the **Multifactor Authentication** dropdown list to select the desired authentication method.
14. Click the **Validate & Save** button. The information is validated. If there are any connectivity issues, an error message will appear stating what field is the likely cause. If the Active check box is not selected no validation occurs. If you chose anonymous authentication, no secret is needed and no credential validation occurs; however the distinguished name and FQDN are still used. Upon a successful save, a new box appears, prompting the user to select their initial synchronization groups. If groups appear in the search box that also indicates the connection was successful.

Secure LDAP

Overview

By default, Secret Server uses normal LDAP on port 389 to communicate with Active Directory. Although passwords are still transmitted using Kerberos or NTLM, user and group names are transmitted in clear text. In contrast, secure LDAP (LDAPS) requires that both port 389 and 636 are open.

If you want all information to be encrypted, then you can enable Secure LDAP (LDAPS) in Secret Server via the Advanced link on the Edit Domain page.

When LDAPS is used, Secret Server transmits and receives Active Directory data through port 636 (with port 389 open). A certificate on the domain controller is used to negotiate encryption, and no information is transmitted in clear text.

> If you want to use Integrated Windows Authentication and Secure LDAP, that is only supported in Windows Server 2008 R2 or greater.

Troubleshooting LDAPS Connection Issues

Common problems with LDAPS and Secret Server:

- When you turn on LDAPS you will get a "domain name is invalid" error.
- Users are suddenly unable to log on Secret Server.

Both issues are caused by LDAPS to Secret Server communication issues, usually one of the following:

- The certificate is expired (this is the client certificate, not the SSL on the Secret Server website).
- LDAPS is not enabled in your environment.
- A port is blocked that is denying successful communication between the server and AD.

To troubleshoot, use the free LDP tool to test LDAPS connections from the Secret Server Windows server to your AD server. If you are unable to establish a connection on port 636 (with 389 open too), then we recommend consulting with your AD or security team.

> Sometimes the Secret Server event viewer has information regarding invalid certificates.

Secret Server Discovery

Secret Server Discovery is a powerful feature that scans an environment to identify accounts and associated resources, known as dependencies. This process helps in creating new secrets within Secret Server by automatically discovering and importing accounts, thereby reducing the administrative burden of manually tracking computers and accounts. Discovery can find various types of accounts, including Windows local admin, Windows domain, and Unix non-daemon accounts, as well as dependencies like scheduled tasks, application pools, and services running under domain accounts. It supports multiple discovery types, including Active Directory, ESX/ESXi, AWS, Google Cloud Platform, and Unix. Additionally, Secret Server allows for extensible discovery through custom PowerShell scripts, enabling the discovery of account and dependency types not supported out-of-the-box. This feature ensures continuous monitoring and management of privileged accounts, enhancing security by identifying and securing backdoor accounts and other potential vulnerabilities.

For details, refer to the "Discovery Overview" on the next page.

Understanding Discovery

"Discovery Overview" on the next page

"Discovery Glossary" on page 525

"Introduction to Discovery Sources, Scanners, and Templates" on page 525

Discovery Overview

Discovery is the process where Secret Server scans an environment to find accounts and associated resources called *dependencies*. Once accounts are found, they can be used to create new secrets in Secret Server. Users with the "administer discovery" role permission can either manually import accounts or can create an automated process to do so. Using discovery does not stop users from manually creating their own secrets.

Some typical accounts that discovery can find include Windows local admin, Windows domain, and Unix non-daemon. Some typical dependencies discovery can scan for include scheduled tasks running as a domain user, application pools running as a domain user, and services running as a domain user.

> Account and dependency types not supported out-of-the-box in Secret Server can still be discovered by writing PowerShell scripts that you can run as custom scanners. See "Extensible Discovery" on page 571.

In a Hurry?

We suggest reading (in order):

- "Discovery Glossary" on page 525
- "Introduction to Discovery Sources, Scanners, and Templates" on page 525
- "Running and Interpreting Active Directory Discovery" on page 596

Discovery Benefits

Importation of Network Credentials

By using discovery, your Secret Server offsets the burden of keeping track of computers and accounts on your network. This can be especially beneficial when getting started for discovering and importing accounts in bulk, as well as having Secret Server find accounts and create secrets whenever a new machine or account is provisioned.

Protection Against Backdoor Accounts

When Secret Server is configured to discover new accounts, it provides added protection by regularly running discovery on your network to identify those accounts. Secret Server adds the new accounts to its records and resets the accounts password to values that meet your security policy. Consequentially, if someone is setting up backdoor admin accounts on the network, they cannot use those accounts very long before they are imported into Secret Server and their passwords are changed with Remote Password Changing (RPC).

Discovery Types

Active Directory Discovery

Secret Server AD discovery scans for AD machines, AD user accounts, local Windows accounts, and dependencies on an AD domain. First, SS discovers machines from your domain. Next, SS scans each machine for

local Windows accounts and dependencies **that depend on domain accounts**. By default, SS scans for local accounts, domain accounts, scheduled tasks, Windows services, and IIS application pools. You can discover additional accounts and dependencies by creating PowerShell scanners. PowerShell scanners are an advanced topic described in the "Extensible Discovery" on page 571 section.

ESX/ESXi Discovery

Secret Server provides a wizard to help configure ESX/ESXi discovery. You name the discovery Source, define the host ranges of the desired IP addresses, and choose a secret to use as credentials when scanning.

> Secret Server provides a "Generic Discovery–Only Credentials" secret type that stores a simple username and password pair for Unix or ESX/ESXi discovery. It is intended only for discovery and is incapable of RPC.

AWS Discovery

Secret Server can scan Amazon Web Services (AWS) for accounts that can access the cloud resource. Two types of secrets can be discovered and managed through Secret Server:

- AWS Access Key: Keys used for programmatic integration with AWS.
- AWS Console Account: User login accounts for AWS.

Google Cloud Platform Discovery

Secret Server can manage Google Cloud Platform (GCP) service accounts and VM instances. This feature allows users to run discovery to pull and manage VM Instances, as well as import and manage GCP service accounts.

Unix Discovery

Secret Server provides a wizard to help configure Unix discovery. You name the discovery Source, define the host ranges of the desired IP addresses, and choose a secret to use as credentials when scanning. The default command sets that Secret Server ships with discovers machines and accounts in most Unix environments.

By default, the "Find Non-Daemon Users (Basic Unix)**"** command set is used first. If a built-in account is discovered, you must modify the discovery source to use the "Find All Users (Basic Unix)" command set. You can create new command sets by clicking the Configure tab on the Discovery Sources page.

Extensible Discovery

You can customize discovery by changing parts of it to use PowerShell. The information a discovery scanner outputs is defined by its scanner template. For standard templates, the input and output information types are fixed. Extensible discovery allows you to customize or replace the unmanaged account, IP address and OU, account, and dependency discovery steps above. Extensible discovery does still have limitations on what information is passed between discovery scanners. For more information, see "Extensible Discovery" on page 571.

Discovery Performance

Please see our "Discovery Best Practices" on page 550 to learn about optimizing discovery performance.

Example Discovery Process

A typical automated discovery process for Active Directory domains, running on an interval, looks like this:

> The majority of current discovery processes are for AD discovery source type. The others types differ by input and output but follow a similar process.

> Even though automatic discoveries run on a set interval, you cannot schedule when those occur. The interval is from whenever the discovery last ran.

1. Discovery matching runs. The discovery matcher creates a link between existing active secrets and any existing secrets in Secret Server based on their machine names, accounts and dependencies. The matcher is automatic. When matches are found, the corresponding existing discovery results appear as "managed" in the discovery network view with a link to the existing secret or dependency.
2. Discovery rules run and attempt to match any unmanaged discovery results to the rule's parameters. If a rule matches the results, discovery automatically imports the results using the settings in the discovery rule. Once finished, discovery begins.
3. The Find Host Ranges scanner (using the Windows Discovery base scanner) runs with an Active Directory Domain input template. The scanner determines which OUs are to be scanned and populates its Organizational Unit output template with a list of those OUs. The output template will be used by the following Find Machine scanner and also by the Find Local Accounts scanner, which does not require machine information.
4. The Find Machine scanner (using the Windows Discovery base scanner) examines OUs from its Organizational Unit input template via LDAP and creates a list of machines with which it populates its Windows Computer output template. This is the list of computers to run a dependency scan on. The Find Dependencies scanner uses this instance of the output template as its input template.
5. The Find Local Accounts scanner (using the File Load Discovery base scanner) examines OUs from its Organizational Unit input template via LDAP and creates a list of all AD admin accounts with which it populates its Active Directory Account output template. This is the list of discovered admin accounts.
6. The Find Dependencies scanner (using the Windows Discovery base scanner) examines a list of machines from its Windows computer input template using various technologies. For example, application pools use Microsoft Web Administration (WMA) or, failing that, Windows Management Instrumentation (WMI). Services use WMI, and scheduled tasks use Windows' task scheduler interfaces. The Find Dependencies scanner can return any number of output templates as desired. These include: Com+ Application, Computer Dependency (Basic), PS Dependency, Remote File, SQL Dependency (Basic), SSH Dependency (Basic), SSH Key Rotation Dependency, Windows Application Pool, Windows Scheduled Task, and Windows Service.

The discovered dependencies for local accounts are displayed at **Admin > Discovery > Discovery Network View > Local Accounts**. Returned accounts for AD users are displayed at **Admin > Discovery > Discovery Network View > Domain > Cloud Accounts**.

> Any dependencies that were discovered in prior discovery runs that are no longer present are removed from the discovery results, and their secret dependencies are deactivated.

Manual Discovery

You can also run discovery manually by going to **Admin > Discovery** and clicking the **Run Now** button and selecting **Discovery Scan**. We recommend waiting for any automatic discovery to idle before starting a manual discovery run. A discovery scan runs the first four of the automated steps above. When you click the "Run Now" button on the Scan Computers tab, the last two are run. These steps are the most time intensive steps because many machines may be scanned.

Discovery Glossary

- Command sets: An SSH script that runs on Unix machines and produces a specific set of output to be consumed in a discovery source flow.
- Discovery scan template: A scan template simply defines an object and what properties the object contains. For example, a computer account has a name, machine, and domain. Think of a scan template as an interface that describes an object.
- Discovery scanner: This item defines how to take that information and runs code to produce collection outputs. Scanners can be system out-of-the-box code that runs natively in the system or completely custom scripts that can do anything.
- Discovery scripts: In the scripting section, you can define a script for a discovery scanner. While scripts are not specific to discovery, they are an important piece to help use the power of extensible discovery.
- Discovery source flow: A collection of scanners that work in a common pipe and filter architecture where each scanner inputs a certain type of item and then outputs a different type of item. For example, a scanner takes an input of a host IP range and outputs multiple computers that can then be consumed by another scanner which can input computer information and output computer accounts.
- Discovery source: This defines the definition for how items are discovered. One discovery source may discover Active Directory items, and one may discover Unix machines. It is common to have multiple discovery sources. Each source defines credentials, scanners, and settings specific to your network.
- Secret search filters: Certain scanners and import rules can leverage a filter that uses the name of the machine to find or use an associated Secret. For example, you may have a pattern of naming the local account on a machine including the machine name. A secret search filter allows you to find secrets using the name of the current machine in the pattern to find the matching secret.
- Discovery Import: Import performs the task of creating new Secrets or linking dependencies to existing Secrets based on discovered account objects. It is available as both a manual option and an automated process (dependent on licensed features).

Introduction to Discovery Sources, Scanners, and Templates

Discovery Source

A *discovery source* is a named collective, ordered system that conducts discovery. There are five broad types: Active Directory, Amazon Web Services, Unix, VMware ESX\ESXi, and Google Cloud Platform.

Configuring discovery is defining the parameters of the discovery source, once the general type is chosen.

Each discovery source is a configurable definition of how to scan for computer assets in a given environment. A subcomponent of discovery source, called a scanner, details how to perform those scans.

Discovery Scanner

A *discovery scanner* is a component of a discovery source that collects information during a discovery. There are four general types of scanners, called *scan templates* (in their sequential running order):

- Find host ranges
- Find machines
- Find (local) accounts
- Find dependencies

> They are called scan *templates* because when you create an instance a discovery source, it includes scanners based on a standardized set scanners specific to the platform the discovery source is designed for and the type of performed scan. That is, when you create a discovery source, you are instantiating a set of of scanner objects copied from a set of static templates. You cannot modify the templates, but you can modify the scanners based on them.

Thus, a discovery source consists of an ordered sequence of discovery scanners, along with some data specific to the whole discovery source. Each scanner has a defined input and output, which are also based on object templates. A discovery source can have more than one scanner of a given type.

Discovery Input Template

The defined input type for a discovery scanner. An instance of the template contains the data needed to conduct the scan. The input template is often, but not always, an output template of the preceding scanner in the sequence. Some examples include Active Directory domain, AWS discovery source, organizational unit, and Windows computer.

Discovery Output Template

The defined output type for a discovery scanner. An instance of the template contains the data produced by the scan. The output template is often, but not always, an input template of the next scanner in the chain. Other times, the output may be used by another non-adjacent scanner in the discovery source.

You can also have more than one scanner of the same type in the same discovery source. For example, you could have both the Windows Local Accounts and Active Directory User Accounts scanner active in the Find Accounts section. Click the **+** icon next to the scanner section to see what other scanners are available there.

Some examples include: Active Directory account, AWS access key, ESXi local account, host range, organizational unit, and Windows local account.

Example

The following figure shows the data flow through the discovery source as the scanners receive and output data via input and output templates. The dataflow is as follows:

1. The original input, the domain, comes from the discovery source and was manually inputted.
2. The Active Directory Organizational Units (of type Find Host Ranges) scanner receives the domain via the Active Directory Domain input template.

3. The scan discovers the OUs of the inputted domain and returns those OUs via the Organizational Unit output template.
4. The Active Directory Computers (of type Find Machines) scanner receives the OUs from the Organizational Unit output template.
5. The scanner discovers the Windows computers belonging to the inputted OUs and returns those Windows computers via the Windows Computer output template.
6. The Windows Local Accounts (of type Find Accounts) scanner receives the Windows computers from the Windows Computer output template.
7. The scanner discovers local accounts belonging to the inputted Windows computers and returns those local accounts via the Windows Local Account output template to Secret Server*and* to the Windows Service scanner.
8. The Windows Service (of type Find Dependencies) scanner receives the Windows computers from the Windows Computer output template. This is the same input received in the last step by SS.
9. The scanner discovers Windows services belonging to the inputted Windows computers and returns those Windows services via the Windows Service output template to SS.

Editing and Adding Discovery Scanners

Many of the discovery scanners can be edited after instantiation by the discovery source. Sometimes the editable data is the same as was originally inputted for the discovery, and sometimes it is something else altogether.

The Secret Credentials section is the same as the credential secret defined when creating the discovery source. The Advanced Settings section contains settings that were not initially configurable in the discovery scanner.

In this case, you see many default configuration settings that were not originally settable in the discovery source. For example, you can use a different credential secret or change the ports scanned by the scanner. Please be advised that if a privileged account is set under specific OUs, that the account will override any accounts set on the individual Discovery Scanners (Service, Scheduled Task, Local User, etc.) As mentioned above, you can also add entire scanners too—that is, more than one scanner of the same type.

General Topics

- "Enabling Specific OU Domain Discovery" on page 568
- "Extensible Discovery" on page 571
- "Manually Importing Local Accounts" on page 592

Account Permissions for Discovery

Entra ID

The Application Registration (mapped to an Azure Application Registration Secret) used for Entra ID Discovery must be assigned to the following roles:

- EntitlementManagement.Read.All
- RoleManagement.Read.Directory
- User.Read.All

> Permissions must be added as Application Permissions not Delegated Permissions.

Unix

Local Accounts

The scanning account needs to be able to connect over SSH and read the contents of `/etc/passwd`. If Discovery needs to take over an account then the scanning account will also need the ability to run `sudo passwd <username>`

> During a remote password change, the `passwd` command is also used and the default local settings will apply.

SSH Public Keys

The scanning account needs the ability to login and execute `sudo` without a password prompt.

> Please see "Discovering SSH Public Keys" on page 636 for more information.

ESXi

The scanning account needs **Shell Access** and the **Query VRM Policy** permission.

Local Windows Accounts

The scanning account needs the **Access this computer from the network** permission (and possibly one more) on the endpoint:

1. Access the windows command line and run `gpedit.msc`. The **Local Group Policy Editor** window will open.
2. Go to **Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.
3. Double-click the **Access this computer from the network** policy. The properties for the policy appear.

4. Ensure the scanning account is one of the listed users. If not, click the **Add User or Group** button to add it.

> Modifying this policy may overwrite or remove access to the device for remote processes. This policy is not usually configured by default, so any existing inherited permissions could be overwritten.

5. Look at the following list of operating systems and updates to determine if any of them match your system:
   - Windows 10, version 1607 and later
   - Windows 10, version 1511 with KB 4103198 installed
   - Windows 10, version 1507 with KB 4012606 installed
   - Windows 8.1 with KB 4102219 installed
   - Windows 7 with KB 4012218 installed
   - Windows Server 2019
   - Windows Server 2016
   - Windows Server 2012 R2 with KB 4012219 installed
   - Windows Server 2012 with KB 4012220 installed
   - Windows Server 2008 R2 with KB 4012218 installed

> For more information on this security issue, see Network access: Restrict clients allowed to make remote calls to SAM.

6. If you found a match, do the following:
   a. Go to **Computer Configuration > Windows Settings > Security Settings > Local Policies > Security Options**.
   b. Double-click the **Network access: Restrict clients allowed to make remote calls to SAM** policy. The policy properties appear.
   c. Click the **Edit Security** button to select an account for the Security descriptor text box. The **Security Setting for Remote Access to SAM** dialog box appears.
   d. Ensure the scanning account is present (if not add it).
   e. Click the account in the **Group or user names** list. The permissions for that account appear.
   f. Ensure the **Allow** checkbox next to the **Remote Access** permission is selected.
   g. Click the **OK** button.

> The discovery account must be a part of the local admin's group to be able to pull back any local accounts.

Windows Services, Scheduled Tasks, App Pools, and COM+ Applications

> There are special considerations for discovering service accounts running COM+ Applications, please see "Windows Services" on page 934.

To scan for service accounts, the account entered must be a domain account that is in the Administrators group on the target machine(s). Follow the instructions below in either case to ensure your account has the appropriate privileges to run a successful scan:

1. Access the windows command line and run `gpedit.msc`. The **Local Group Policy Editor** window will open.
2. Go to **Computer Configuration > Preferences > Control Panel Settings**.
3. Right-click **Local Users and groups** and select **New > Local Group**.
4. Leave the **Action** dropdown list set to **Update**.
5. Click to select **Administrators (Built-in)** in the **Group Members** drop-down list.
6. Click the **Add...** button.
7. Search for the account you will use for Discovery scanning.
8. Click the **OK** button to save your changes.

   The next time the group policy updates across your environment, the discovery account will be a part of the local administrators group.
9. For stronger security, we suggest configuring the group policy to limit the login privileges of the Discovery account:

   > This will also require you to use a different account (separate from the discovery account) to rotate dependencies.

   a. Access the windows command line and run `gpedit.msc`. The **Local Group Policy Editor** window will open.
   b. For your domain policy, go to **Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.
   c. Add your discovery account to the **Deny log on locally** policy.
   d. Add your discovery account to the **Deny log on through Remote Desktop Services** policy.
   e. (Optional) Ensure the account is not part of the remote desktop users group.
10. Do not put dependency changers in these policies:
    - Deny access to this computer from the network.
    - Deny log on Locally.
11. We recommend putting dependency changers in these policies:
    - Deny log on as a batch job.
    - Deny log on as a service.
    - Deny log on through Remote Desktop Services.

Application Pool Discovery Over Distributed Engines

Application pools cannot directly perform discovery over distributed engines, but a work around can be made by using impersonation of a discovery source's identity.

> The distributed engine service must be run as an account with sufficient privileges to scan the application pools on its own.

For discovery to work on application pool service accounts on IIS 7+, the **IIS Management Scripts and Tools** feature must be installed on both the DE and the web server running the scan and the target machine hosting the application pool.

Check that the following privileges are set:

1. Enable the account to log in as a service.
2. Grant the account read, write, and execute privileges to the entire distributed engine installation directory and sub-folders.
3. Add the account to the administrators group on each computer that will be scanned.

80070005 Error

An 80070005 error indicates that the distributed engine service account does not have sufficient privileges on the machine being scanned:

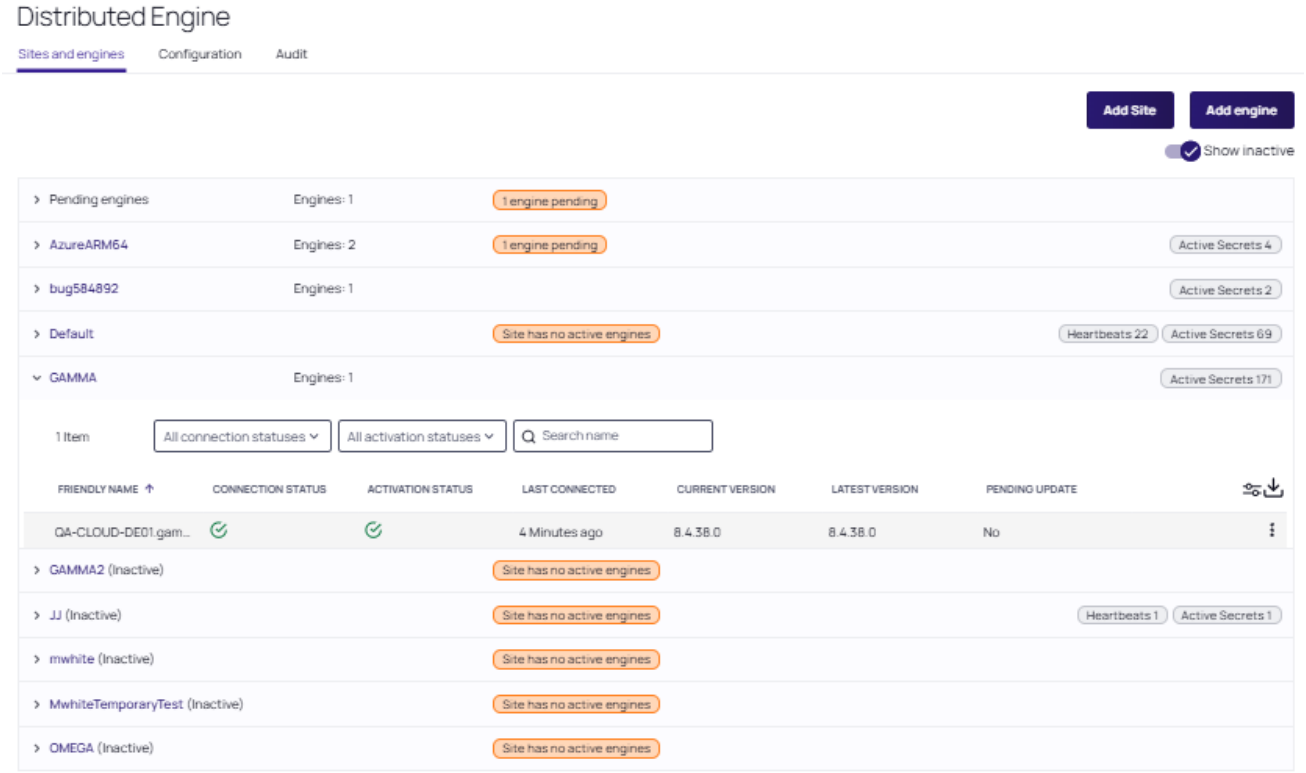
```
Exception: Retrieving the COM class factory for remote component with CLSID {2B72133B-3F5B-4602-8952-803546CE3344} from machine <MACHINE> failed due to the following error: 80070005
```

To resolve the error, add the account to the administrators group on that machine. When changing the user of the distributed engine service, Secret Server interprets the service as a new engine. Delete the old engine and activate the new engine in secret-server.

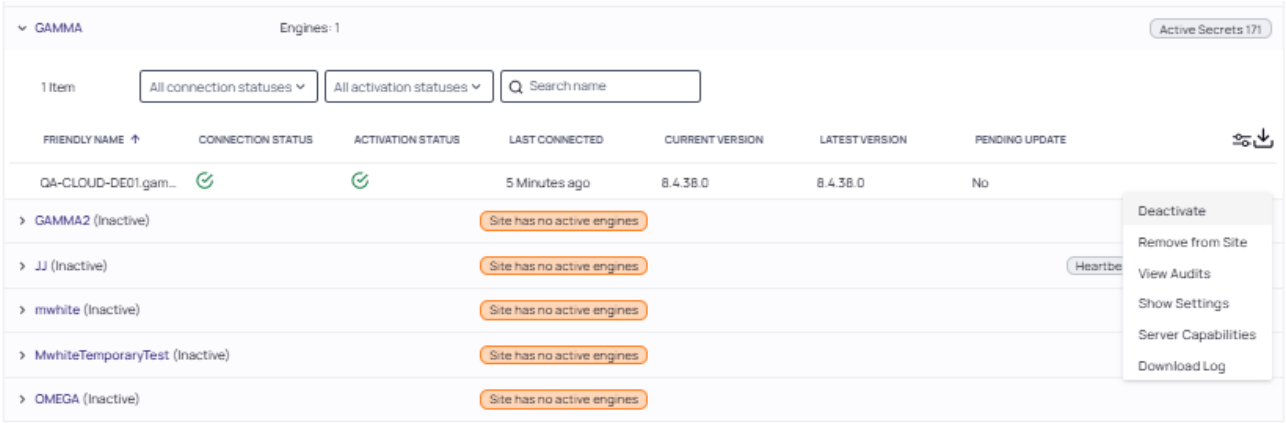
The typical workflow for changing the account login for the distributed engine service is as follows:

1. Hover over **Settings** and under **Distributed Engine** select **Sites and engines**. The **Distributed Engine** page appears.

2. From the list of sites with active engines displayed by default, select the site whose engine you want to deactivate and click the arrow to expand the selection:



3. To deactivate the DE, hover over the far right of the screen on the line of the engine under the download button, a vertical 3-dot option will appear. Click the option to open a drop-down list and select the first option:



4. When prompted, confirm the deactivation by clicking **Ok**.
5. Delete the engine by reselecting the 3-dot menu and choosing **Remove from Site**.
6. When prompted to confirm this action click **OK**. This will place the engine under the **Pending engines** dropdown at the top of the page. In order to see it in that section a refresh of the page is necessary.

7. Once the deactivated engine can be seen in the list, hover over the line again until you see the 3-dot option menu, and click on it. The **Delete** option is second on the list, choose it and confirm the action. The page will refresh automatically and the engine will vanish from the **Pending engines** list.
8. Go to the distributed engine server and stop the distributed engine service or end the process from the Task Manager.
9. Change the Log On option – this must be the same account used for your discovery.
10. Start the distributed engine service.
11. In the Secret Server application, go to **Settings > Distributed Engine > Sites and engines** and check if the engine is under **Pending engines**.
12. If so, activate the DE by clicking the 3-dot menu and selecting **Activate** and assigning it to a site when prompted. You should see a successful connection. **Activation Status** should also have the green checkmark.

Managing IIS Application Pool Dependencies

To manage IIS Application Pools effectively, the same requirements that apply to Discovery must also be met. This includes ensuring that the necessary permissions and configurations are in place, even for customers who create dependencies manually. See Application Pool Discovery Over Distributed Engines for more details.

To manage IIS dependencies, the Distributed Engine service must be run as a service account with administrator rights on the target workstations. This is crucial because:

- Administrator rights ensure that the service account can access and manage all necessary components of the IIS Application Pools.
- Without these rights, the service may not function correctly, leading to potential issues in managing dependencies.

Configure the Distributed Engine Service

1. Create a Service Account with administrator rights on the target workstations and the necessary permissions to access IIS components.
2. Configure the Distributed Engine:
   - Open the Distributed Engine configuration settings.
   - Set the service account credentials.
   - Save the configuration and restart the service to apply changes.
3. Check the logs to ensure the service is running without errors.
4. Test the management of IIS Application Pools to confirm successful configuration.

Creating a Discovery Source

Introduction

A *discovery source* is a named collective, ordered system that conducts discovery. There are six broad types to choose from:

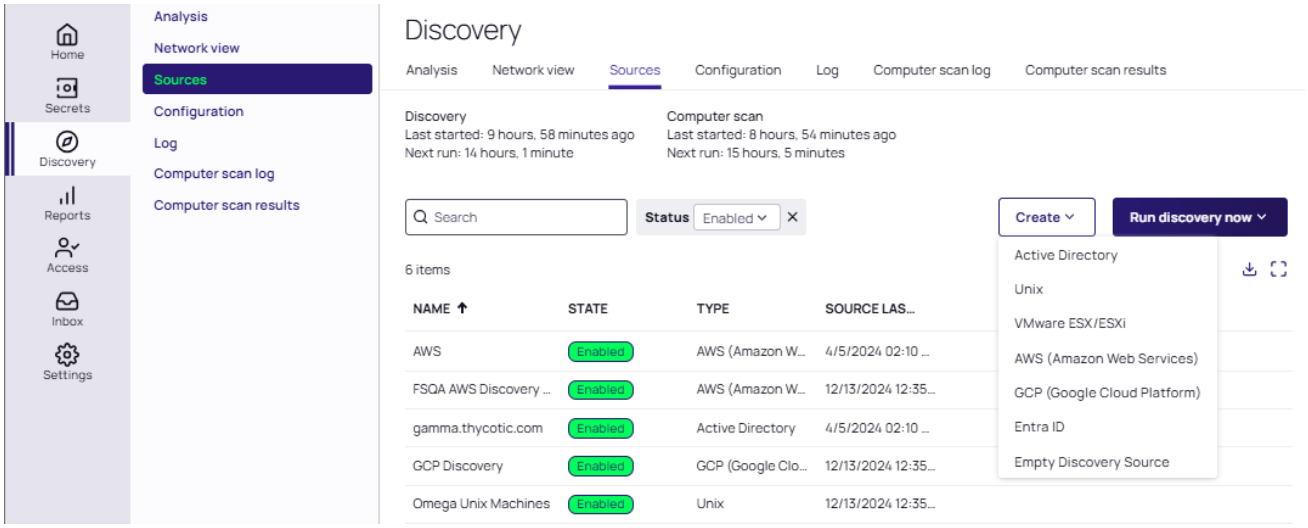
- Active Directory
- Unix
- VMware ESX\ESXi
- AWS (Amazon Web Services)
- GCP (Google Cloud Platform)
- Entra ID

Configuring discovery is defining the parameters of the discovery source. Each discovery source is a configurable definition of how to scan for computer assets in a given environment. A subcomponent of a discovery source, called a scanner, details how to perform those scans.

> There is also an **Empty Discovery Source** option that creates a new discovery source that does not contain any scanners. This is useful for extensible discovery as you can add any custom scanners.

Procedure

1. Search for **Discovery Sources** on the main page and select it.
2. Select the **Create** dropdown list and select the type of source you need:

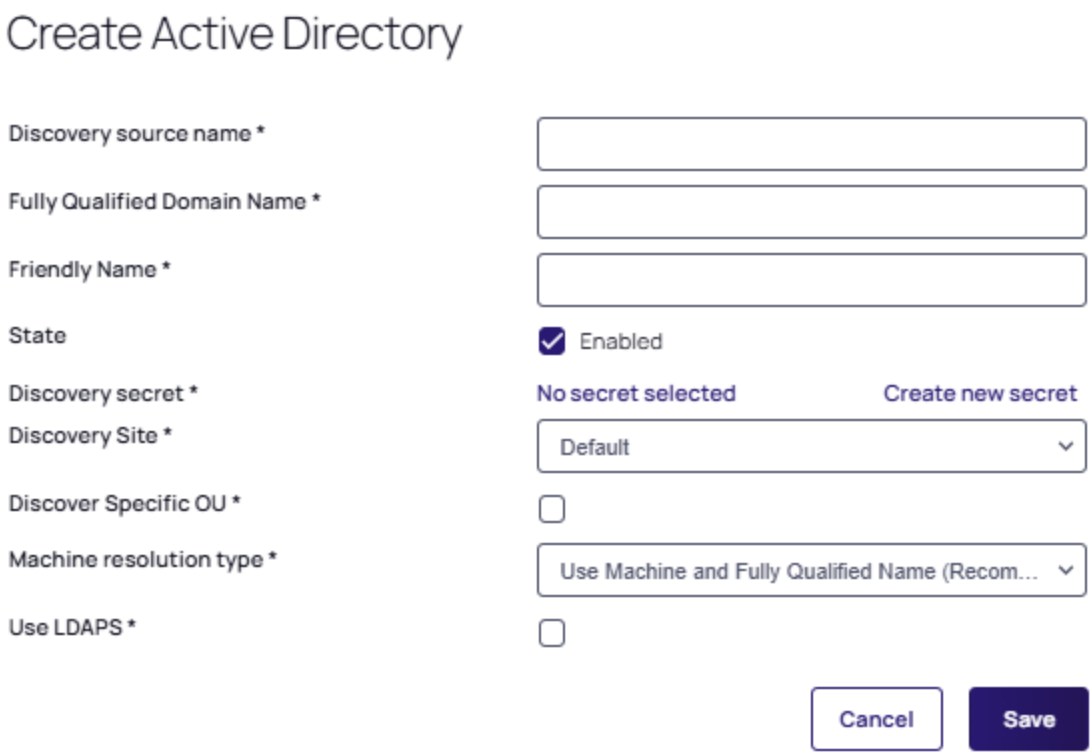


   For all sources except for Active Directory you are prompted for a name, site, source type, and secret. Active directory has a specific dialog which allows for some advanced validation and customization.

Active Directory discovery scans Active Directory (AD) machines, AD user accounts, local Windows accounts, and their dependencies within an AD domain. The discovery process begins by identifying machines within your domain, followed by scanning each machine for local Windows accounts and associated dependencies. By default, the scan includes local accounts, domain accounts, scheduled tasks, Windows services, and IIS application pools. To further enhance the discovery process, you have the option to create PowerShell scanners, which allow for the identification of additional accounts and dependencies. PowerShell scanners are an advanced topic covered in detail within the Extensible Discovery section.

a. If the **Active Directory** option is chosen:

i. The **Create Active Directory** discovery source page appears:



If you upgraded from an earlier version of Secret Server and have created an AD domain within it, a corresponding discovery source is displayed on this page. If discovery was not enabled on that domain, the discovery source Active column is not checked for that discovery source.

ii. Type in values for the following mandatory fields:

- Discovery source name
- Fully Qualified Domain Name
- Friendly Name
- Discovery secret
- Discovery Site
- Discover Specific OU
- Machine resolution type
- Use LDAPS

iii. Ensure the **State** checkbox is set to **Enabled**.

This activates the discovery source for scanning. Enabled discovery sources are scanned at the defined discovery interval. If you have multiple discovery sources, the discovery source with the most un-scanned computers is scanned first.

iv. Click the **No Secret Selected** link. The **Select Secret** popup page appears.

v. Select a secret that will be used as the credentials for discovery scanning and AD synchronization. These credentials must have the proper rights to scan the remote machines.

vi. Once you've selected a secret the popup page closes. The name of the secret you chose replaces the *No Secret Selected* link name.

vii. Alternatively, you can create a new secret for the credentials:

i. Click the **Create New Secret** link. The **Create New Secret** popup dialog appears.

ii. Search for and select the **Generic Discovery Credentials** secret template. Another Create New Secret popup appears.

iii. Type or select the parameters needed for the discovery operation. Parameters with asterisks are required.

iv. Click the **Create Secret** button.

viii. Click the **Discovery Site** dropdown list to select the desired site for the discovery source.

If distributed engines are set up, the list shows all active sites. If no distributed engines are set up, the list defaults to local, and you cannot change it.

ix. Select the **Discover Specific OU** checkbox to limit your discovery to an OU.

See "Enabling Specific OU Domain Discovery" on page 568 to define the scanned OU. When you select this option, a **Domain Scope** tab appears on the Discovery Source page for the created AD discovery source.

x. Leave the **Machine Resolution Type** dropdown list set to **Use Machine and Fully Qualified Name** unless you have a specific reason to change it.

xi. Check the **Use LDAPS** checkbox to if you want this server to connect to the LDAP server using SSL.

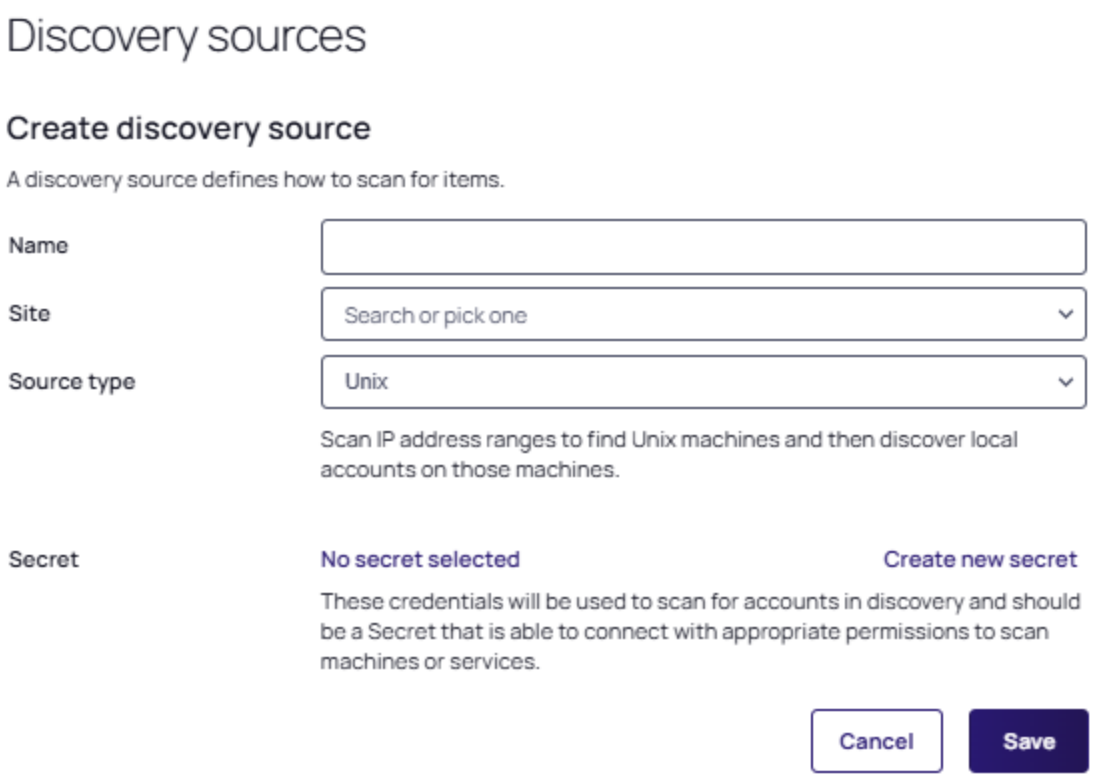
xii. Click the **Save** button.

Secret Server attempts to access the domain with your specified credentials to ensure the configuration is correct. Secret Server must have access to the domain provided, and the account credentials must work.

3. For other source types when selected:

> For Unix, the default command sets efficiently discover machines and accounts in a wide range of Unix environments. By default, the "Find Non-Daemon Users (Basic Unix)" command set is used for discovery. If you wish, however, to include the built-in account in the discovery process, you will need to update the discovery source to use the "Find All Users (Basic Unix)" command set. For further customization, you can create new command sets by accessing the "Configure Command Sets" option on the Discovery Sources list page. Additionally, you can modify the secrets employed during the discovery by accessing the scanner settings.

   a. The discovery sources page appears:



   b. Type the name of the discovery source in the **Name** field.

   c. Click the **Site** dropdown list to select the domain.

   d. Click the **Source Type** dropdown list and choose one of the following:

      - **Empty**: An empty discovery source does not have any scanners in it and after it is created you will need to add scanners before it can be activated. Creating an empty source is for when you have specific

scanners in mind or want to build it from scratch.

- **Unix**: Scan IP address ranges to find Unix machines and then discover local accounts on those machines.
- **AWS**: Scan Amazon Web Services for keys, users, windows and non-windows machines. You will be prompted after saving to select which items.
- **GCP**: Scan Google Cloud Platform for users, windows and non-windows machines. You will be prompted after saving to select items.
- **VMware ESX/ESXi**: Scan IP address ranges to find VMware ESX/ESXi hosts and discover local accounts.
- **Entra ID**: Scan an Entra ID tenant for Roles and Role Members.

e. Click the **No Secret Selected** link. The **Select Secret** popup page appears.

Select a secret that will be used as the credentials for discovery scanning and AD synchronization. These credentials must have the proper rights to scan the remote machines.

Once you have selected a secret the popup page closes. The name of the secret you chose replaces the *No Secret Selected* link name.

i. Alternatively, you can create a new secret for the credentials:

i. Click the **Create New Secret** link. The **Create New Secret** popup dialog appears.

ii. Search for and select the **Generic Discovery Credentials** secret template. Another Create New Secret popup appears.

iii. Type or select the parameters needed for the discovery operation. Parameters with asterisks are required.

iv. Click the **Create Secret** button.

f. Click the **Save** button.

4. The **Add Flow** popup appears. Select the flow that matches the source type you just created.
5. The source details page loads automatically with the **Scanners** tab selected. Make any adjustments to the source scanner flow as needed.
6. All scanners for the source are listed in this tab. Click on one to see its settings in a panel that appears on the right.
7. Select **Edit Scanner** to make any changes to the scanner secret or its IP ranges.

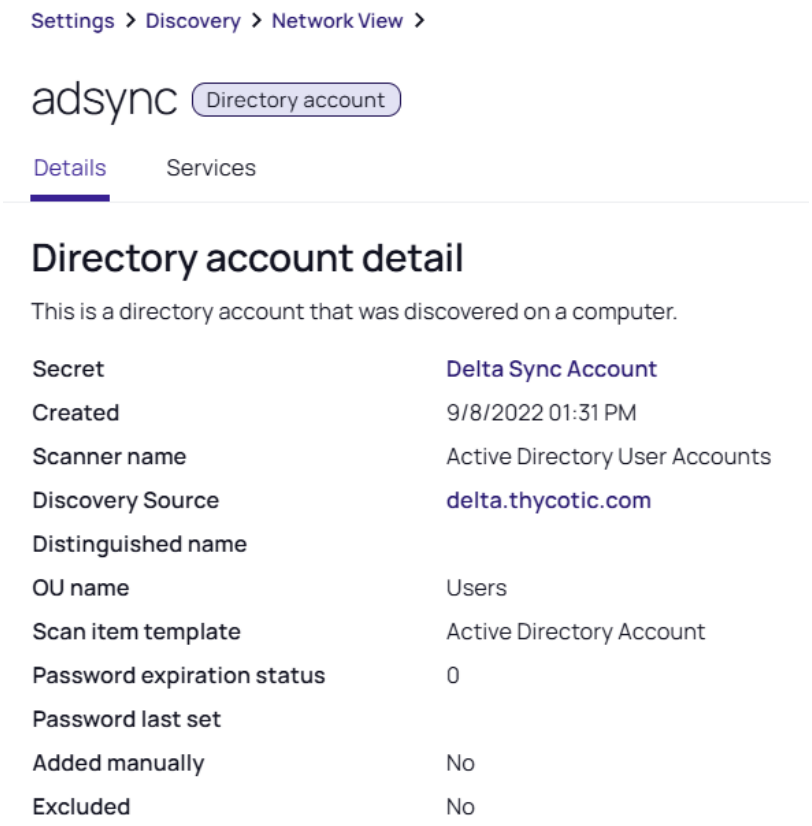
Your discovery source may not be ready to run yet and may require additional properties to be configured for your network. Some scanners will have required properties such as an IP address range and will indicate this by a red exclamation coupled with "Review needed" on the discovery flow on the Scanners tag. Some settings may be specific to your network and require customization. We recommend that you review each scanner and each setting to see which settings apply to you.

Discovery Account Details

To view the Discovery Account Details:

1. Select **Discovery > Network view**.
2. In the **Item Type** dropdown list, select **Directory Accounts**.
3. Click the related account from the list. The account details expands to the right (for a full page view click the **View details** button in the expanded menu).

The account details show the Type, Full Name, Created date and time, Active Directory Account, Delta Sync Account, Discovery Source Name, Managed, Password expiration status. Click **View details** to proceed to the Directory Account details.



Here the following details are displayed:

- **Secret**: refers to the secret associated with the directory account.
- **Created**: the timestamp that indicates when the account was created. It is important for tracking the account lifecycle and understanding its history within the system.
- **Scanner name**: the name of the account as identified during the discovery scan.
- **Discovery Source**: the origin or the source from which the discovery process is initiated. See Discovery Source for more details.
- **Distinguished name**: the unique identifier for the account within a directory service like Active Directory. It provides the full path to the account within the directory hierarchy, which is essential for locating and managing

the account.

- **OU name**: refers to the Organizational Unit (OU) within which the account resides. OUs are used to organize accounts and resources within a directory service, making it easier to apply policies and manage permissions.
- **Scan item template**: defines the criteria and properties used during the discovery scan to identify and categorize the account.
- **Password expiration status**: indicates whether the account's password is set to expire and when.
- **Password last set**: the timestamp that shows when the password for the account was last changed.
- **Added manually**: indicates whether the account was manually added to the system, as opposed to being discovered automatically through a scan. Manually added accounts might require additional verification to ensure they are correctly configured.
- **Excluded**: shows whether the account has been excluded from certain processes or scans. Exclusion might be used to prevent specific accounts from being managed or altered by automated systems, often for security or operational reasons.

Creating Discovery Rules

Introduction

Secret Server discovery rules in play a pivotal role in automating the process of finding, importing, and managing passwords, API keys, and other credentials throughout the IT environment.

Discovery rules offer several advantages:

- Automated Discovery: Discovery rules simplify identifying potential secrets across various platforms and environments, ensuring that no sensitive credentials remain unmanaged or unprotected.
- Policy Enforcement and Risk Reduction: Discovery rules help enforce consistent security policies across an IT environment. This consistency is crucial for minimizing security breaches and ensuring compliance with regulatory standards.
- Efficient Secret Management: Discovery rules reduce the administrative burden on IT teams by automating secret candidate discovery and importation .
- Dynamic Adaptation: Discovery rules help to flexibly adapt to IT environments, which are constantly changing with the addition of devices and applications.

secret-server offers a feature related to discovery rules that allows you to see what accounts would match a rule. This is part of the discovery process where discovery rules are used to automate the identification and management of credentials across various platforms. Here's how it works:

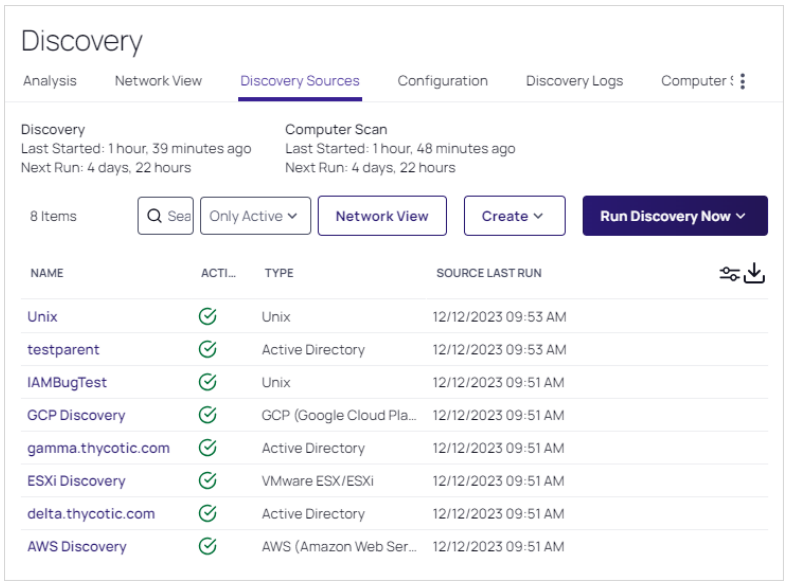
- Discovery Account Rules: These rules are search queries against the accounts found by discovery. When these rules are created and run, accounts that match the rules can be automatically imported as secrets. This means you can define criteria for accounts, and the system will identify which accounts match these criteria.
- Viewing Discovery Results: After running a discovery scan, you can view the results in the Discovery Network View. This view allows you to see which accounts have been discovered and whether they match any existing rules. You can filter and search through these results to identify specific accounts that meet the criteria set by your discovery rules.

Creating Local Account Rules

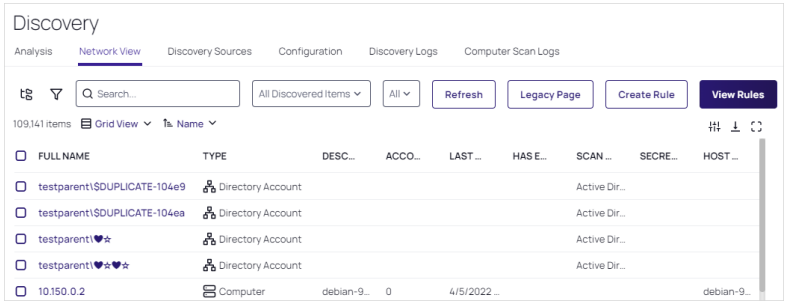
Discovery account rules are search queries against the accounts found by discovery (and visible in the discovery network view). When these rules are created and run, accounts that match rules can be automatically imported as secrets. When matches are found, email notifications can also be sent out. The rule order determines the rule application order. Drag rules to reorder them. Rules can specify a combination of the domain or OU, the computer name and the account name.

To create a rule:

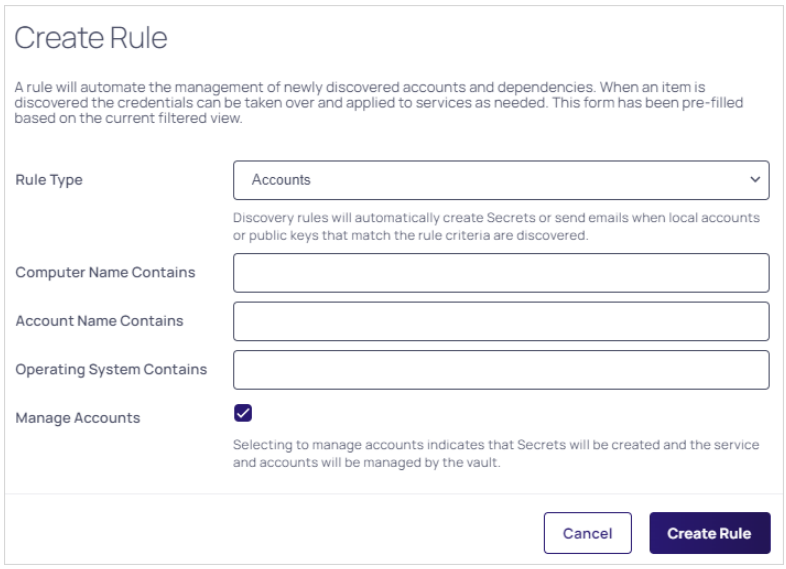
1. Click **Administration > Configuration > General > Discovery**. The Discovery Sources tab of the Discovery page appears:



2. Click the **Network View** tab:

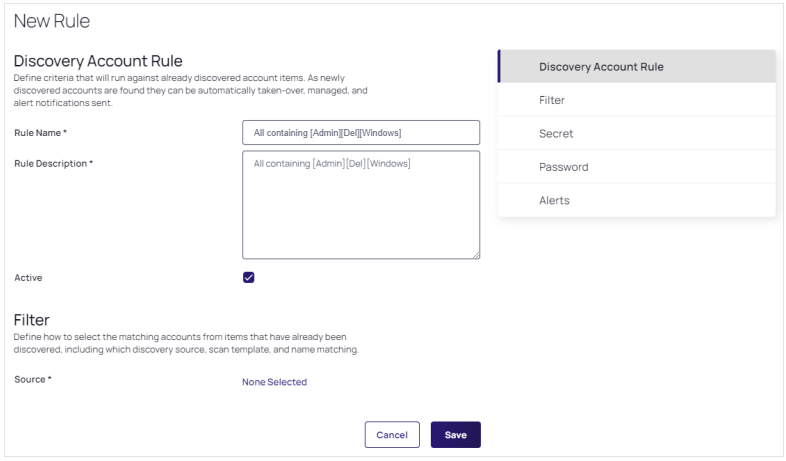


3. Click the **Create Rule** button. The Create Rule popup appears:



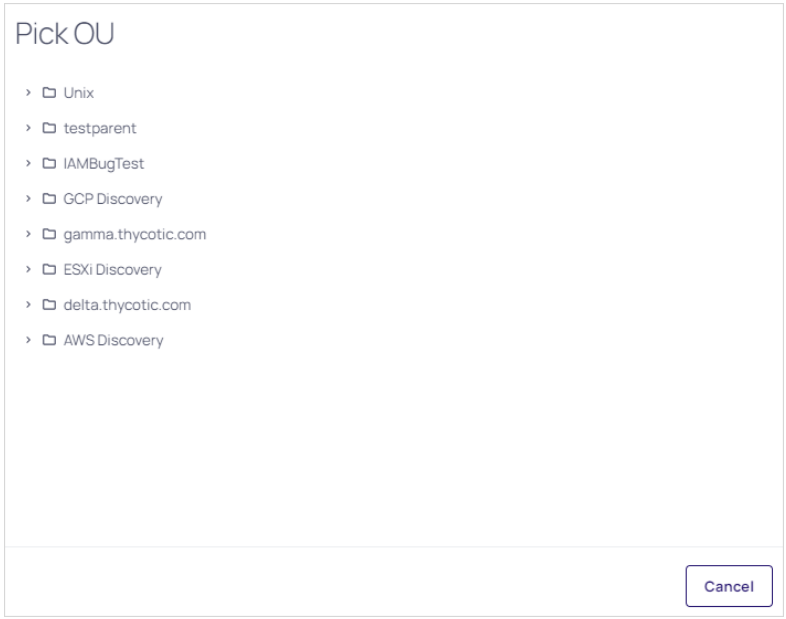
4. Click the **Rule Type** dropdown list and select **Accounts**.
5. (Optional) Type in text strings for the following if you want to limit the scope of the rule:
   - Computer name
   - Account name
   - Operating System

   We typed `Del`, `Admin`, and `Windows`. Discovery rules automatically create secrets or send emails when local accounts or public keys that match the rule criteria are discovered.
6. Click to select the **Manage Accounts** check box if you want secrets to be created and the service and accounts to be managed by Secret Server.
7. Click the **Create Rule** button. A New Rule page appears with the values you typed:

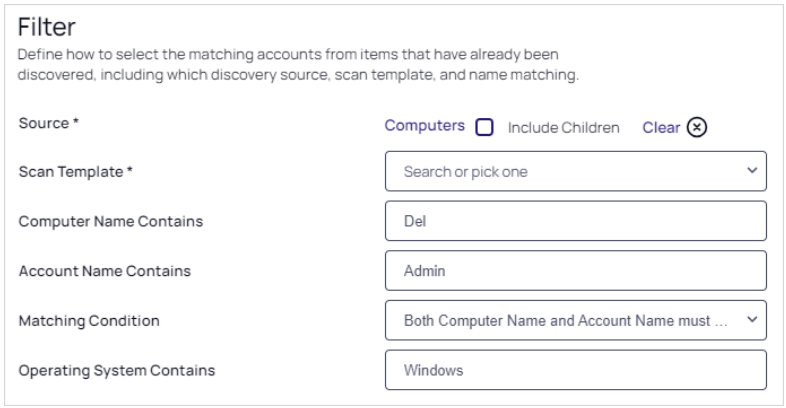


8. Type the name of the new rule in the **Rule Name** text box if you want to change the suggested name.

9. Click the **None Selected** link in the **Filter** section to choose a discovery source. The Pick OU popup appears:



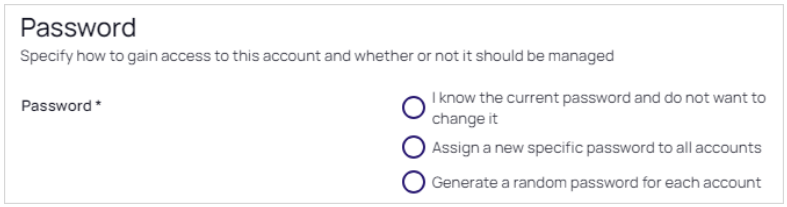
10. Navigate to and select the OU of your choice. Your choice appears as the source, and additional controls appear:



11. Click to select the **Include Children** check box if you want to include any child OUs in the scan.
12. Click the **Scan Template** dropdown list to select an output template.
13. Click the **Matching Condition** dropdown list to pick which of the filtering parameters must match.
14. Note that the completion checklist updates with a check mark to show that you completed the Filter section.
15. Scroll down to the **Secret** section:
16. Click to select the **Create Secrets** check box to enable the section.
17. Click the **Secret Template** dropdown list to select the secret template the new secret will originate from.
18. Click the **Folder** link to select a folder for the new secret to belong to.

> You may not use personal folders for this purpose.

19. Type the naming convention for the new secret in the **Secret Name** text box. We automatically suggest a naming convention based on the hostname and username.
20. Click the **New Secret Permissions** dropdown list to select whether you want secrets to copy (standalone) or inherit (change with the folder) the permissions from the folder.
21. Click the **Site** dropdown list to select the Secret Server local installation or a distributed engine to run the rule from.
22. Note that the completion checklist updates with a check mark to show that you completed the Secret section.
23. Scroll down to the **Password** section:



> Remote password changing must be enabled to change the password.

24. Click to select the **I know the current password…** selection button if you do not want Secret Server to change the account password when the secret is created. Complete the following:
    a. Type the password in the **Current Password** text box.
    b. Leave the **Password Changing** dropdown list set to **Use privileged account**.
    c. Click the **No Secret Selected** link to choose a secret for the privileged account for ongoing use.
25. Click to select the **Assign a new specific password…** selection button if you want all the new secrets to have the same password, which you can later change. Complete the following:

> This option will change the password on the remote machine for any newly discovered accounts.

    a. (Optional) Type a value in the **Take-over Threshold** text box. If the number of accounts that will be taken over exceeds the maximum threshold, the import is canceled and the subscribed users below are notified by email.
    b. Type the new password in the **New Password** text box.
    c. Click the **Password Type** dropdown list to select the desired type.
    d. **Either** click the **No Secret Selected** link to choose a secret for the privileged account for the initial takeover. **Or** if you want to have a set of secrets that can be tried till one works, click the **(Switch to Multiple Reset Secrets)** link and then click the **Add Secret** button to choose the secret. The name of the secret appears. Repeat as needed.

e. Leave the **Password Changing** dropdown list set to **Use privileged account**.

f. Click the **Add Secret** button to choose a secret or secrets for the privileged account for ongoing use.

26. Click to select the **Generate a random password…** selection button if you want to have Secret Server create a strong password for the secret.

> This option will change the password on the remote machine for any newly discovered accounts.

a. (Optional) Type a value in the **Take-over Threshold** text box. If the number of accounts that will be taken over exceeds the maximum threshold, the import is canceled and the subscribed users below are notified by email.

b. Click the **Password Type** dropdown list to select the desired type.

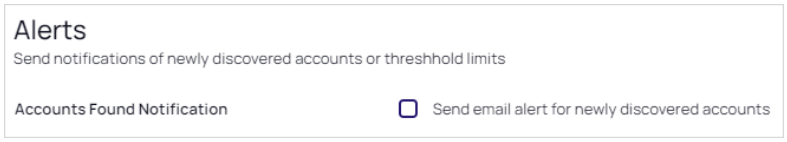
c. Click the **Add Secret** button to choose a secret or secrets for the privileged account for the initial takeover.

d. Leave the **Password Changing** dropdown list set to **Use privileged account**.

e. Click the **Add Secret** button to choose a secret or secrets for the privileged account for ongoing use.

27. Note that the completion checklist updates with a check mark to show that you completed the Password section.

28. Scroll down to the **Alerts** section:

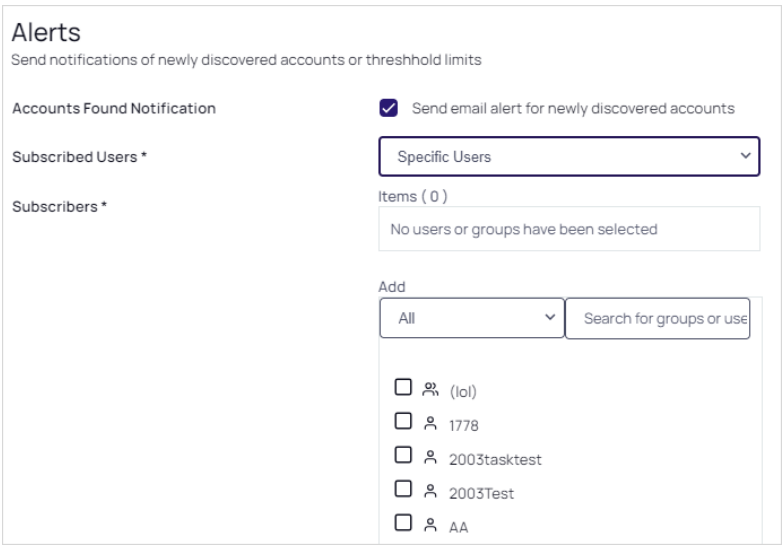


29. Click to select the **Send email alert…** check box to enable the **Subscribed Users** control.

30. Click the Subscribed Users dropdown list to select one of the following:

- **Discovery Administrators** if you only want to notify admins.
- **Specific Users** if you want to define a list of people to notify.

31. If you chose Specific Users, new controls appear:



32. In the **Add** section, select or search for users and groups. As you click each one you desire, it appears in the Items text box.
33. Note that the completion checklist updates with a check mark to show that you completed the Alerts section.
34. Click the **Save** button at the bottom of the page

Creating Dependency Rules

Dependency rules automatically add dependencies (Windows services, schedule tasks, application pools) to existing secrets. You can receive email notifications of linkages by adding an event subscription in the Event Subscriptions page. Rules can specify a combination of the domain or OU.

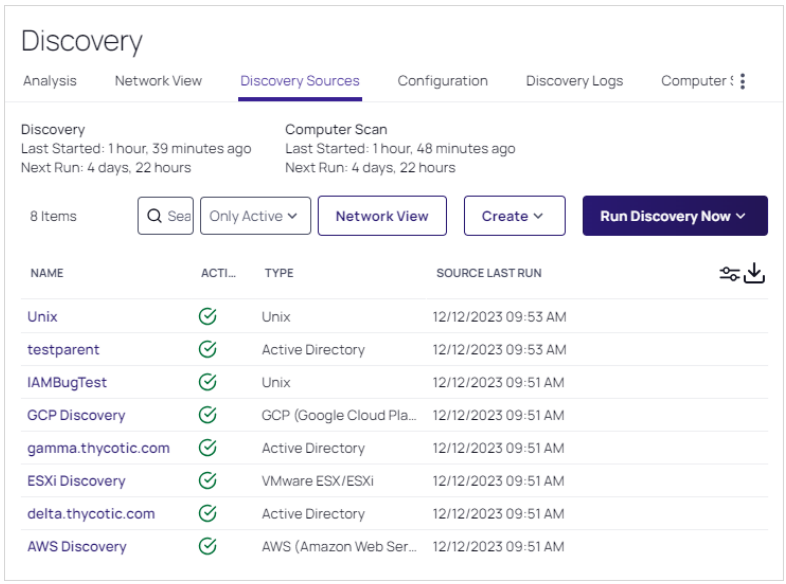
> The rule order determines the order in which the rules are applied. Drag rules to reorder them.

> You must have a discovery scanner and dependency template configured to apply a dependency rule.

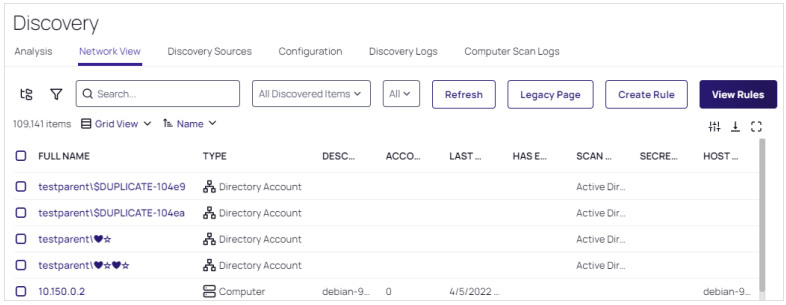
> If you run discovery against Windows Server 2016 or 2019, scheduled tasks are not discovered unless your instance or engine are on the same domain as the target server. On Windows Server 2016 and up, scheduled task discovery only gets a security identifier (SID) for the user that runs the task. Secret Server has code to convert the SID to a username, but this only works if the code is being executed on the same domain as the scheduled task. If the SID cannot be translated, the scheduled task will not be saved with discovery.

To create a rule:

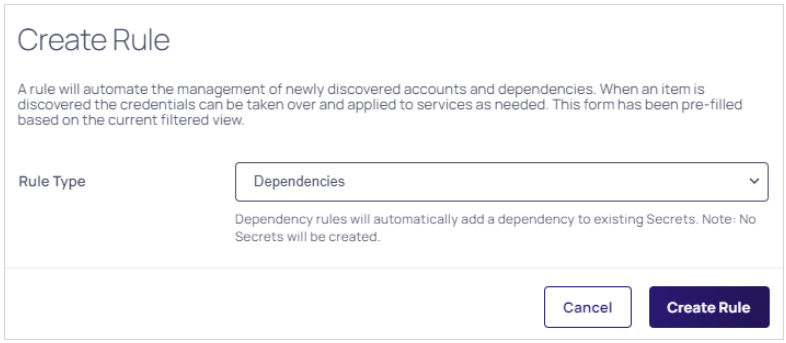
1. Click **Administration > Configuration > General > Discovery**. The Discovery Sources tab of the Discovery page appears:



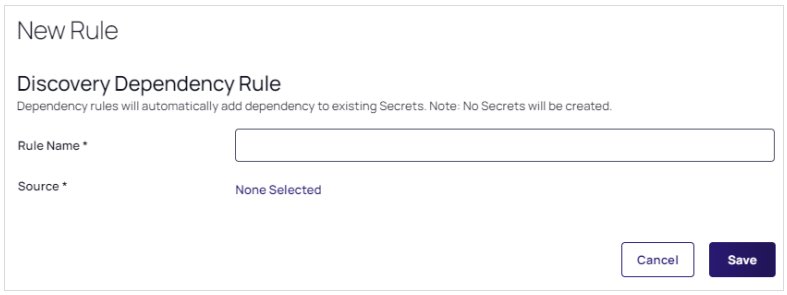
2. Click the **Network View** tab:



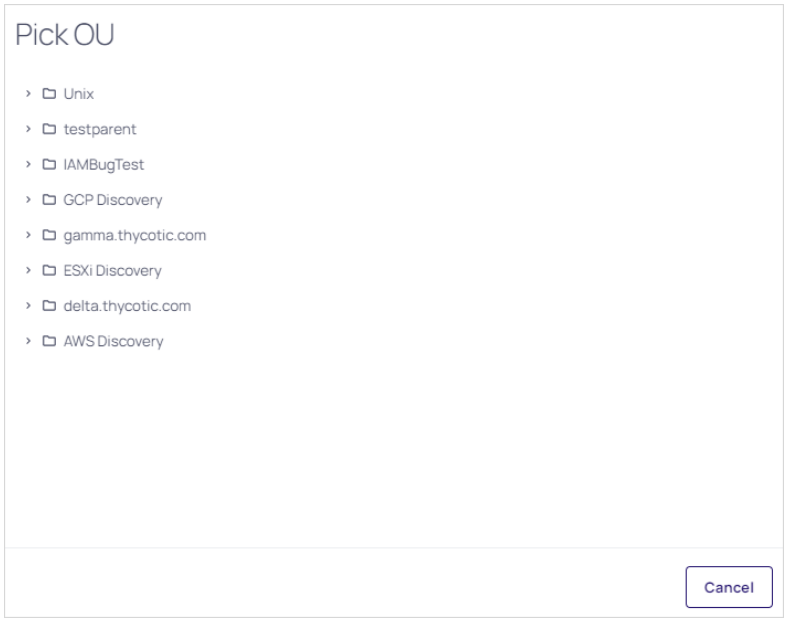
3. Click the **Create Rule** button. The Create Rule popup appears:



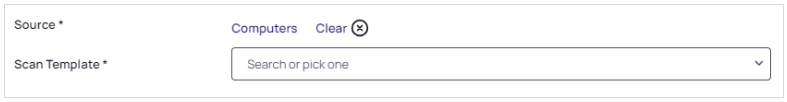
4. Click the **Rule Type** dropdown list and select **Dependencies**.
5. Click the **Create Rule** button. The New Rule page appears:



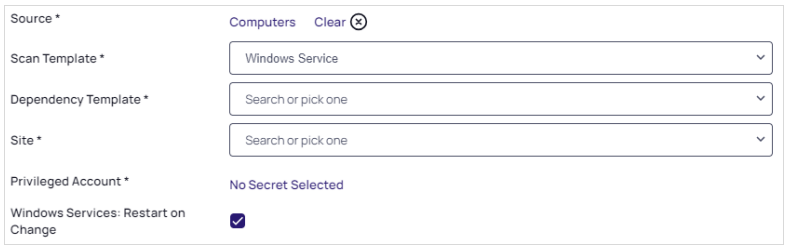
6. Type the name of the new rule in the **Rule Name** text box.
7. Click the **None Selected** link to choose a discovery source. The Pick OU popup appears:



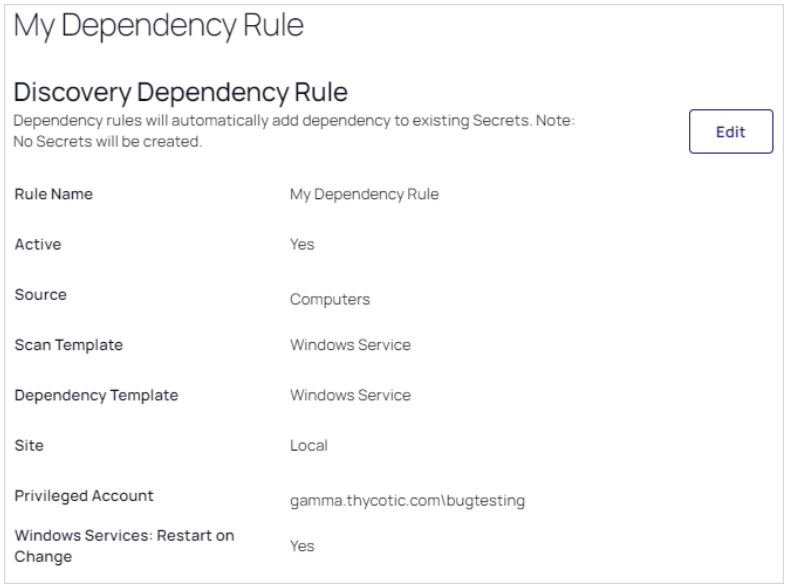
8. Navigate to and select the OU of your choice. Your choice appears as the source, and a Scan Template control appears:



9. Click the **Scan Template** dropdown list to select an output template.
10. Click the **Dependency Template** dropdown list to select a dependency template. For this instruction, we chose Windows Service. Several new controls appear:



11. Click the **Dependency Template** dropdown to select the desired template. Once again, we chose Windows Service.
12. Click the **Site** dropdown list to select the local installation or a distributed engine to run the rule from. We chose Local.
13. Click the **Privileged Account** link to choose a secret for the scanning account. The chosen secret appears as a link.
14. Click to select the **Windows Services: Restart on Change** check box if you want the services restarted after discovery.
15. Click the **Save** button. The page for your new rule appears:



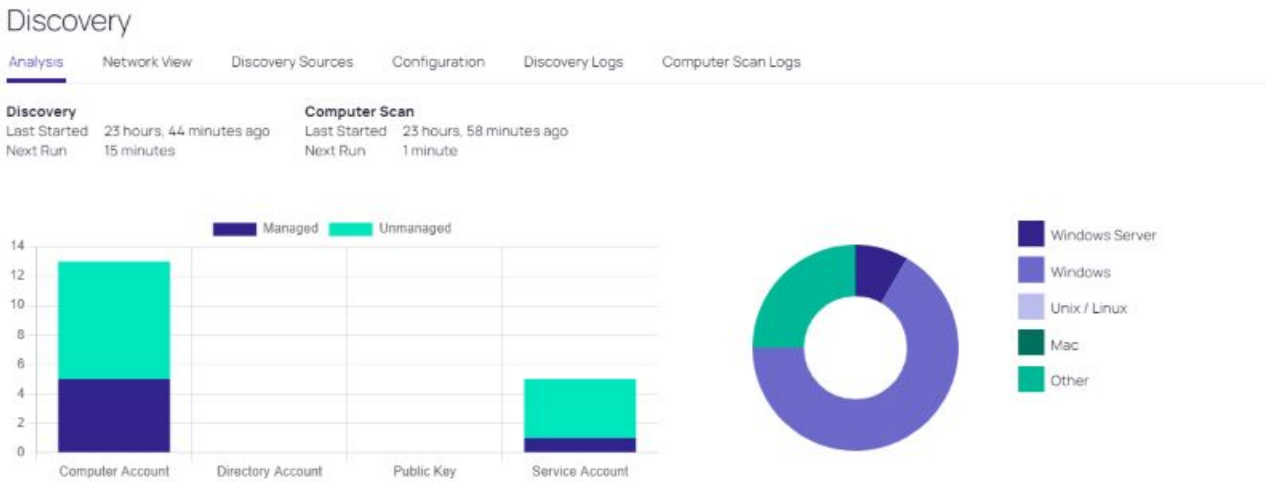
Discovery Analysis

Overview

After discovery has been configured, this page shows high level discovery statistics, including when it ran last and what it has discovered. The chart indicates which items are managed or unmanaged. A *managed* item has a secret associated with it. *Unmanaged* items have been discovered on the network but have not been imported into the vault.

Procedure

1. Click the **Administration** button on the main menu. The All Settings page appears.
2. Click the **Discovery** link in the **Core Actions** section. The Discovery page appears on the Discovery Sources tab.
3. Click the **Analysis** tab. A graphical summary of your discovery activity appears:



TYPE	MANAGED	UNMANAGED
Computer Account	5	8
Directory Account	0	0
Public Key	0	0
Service Account	1	4

OPERATING SYSTEM	TOTAL
Windows Server	1
Windows Server®2008 Enterprise	1
Windows	8
Windows 7 Enterprise	3
Windows Vista™ Enterprise	5
Unix / Linux	0
Mac	0
Other	3
‹script›alert("ComputerVersion");‹/script›	1
XPSP1	1
一个两个三个-ComputerVersion	1

4. Click on an item count link or the column chart to open a network view filtered to only show those items.

Discovery Best Practices

Overview

This document covers the most common settings to tune to make discovery more efficient. Environmental factors contribute to some these settings.

Global Settings

The settings below might make discovery more efficient, regardless an organization's size.

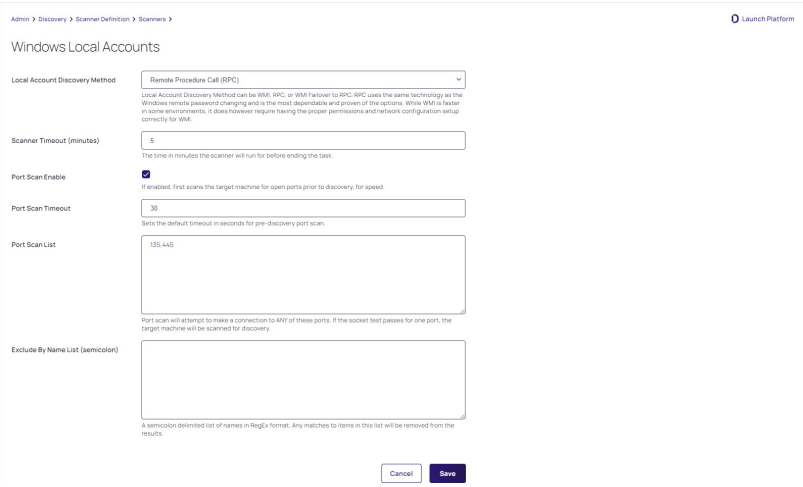
Enabling Port Scanning

> **Note:** During a Port Scan, if the connection attempt fails due to a network failure prior to the port scan timeout another attempt will be made with the remaining time. This means that port scans will run a minimum of the time specified. Also the operation can take up about 20 seconds longer than the port scan timeout to complete. This applies to Secret Server version 11.6 and later.

Introduction

Port scanning is a scan that can be conducted before the regular discovery scan to potentially reduce discovery time—if specified ports are unavailable on a given machine, the standard discovery scan will eventually timeout (the default is five minutes). Port scanning eliminates that timing out process, which saves time.

Figure: Edit Discovery Scanner for Windows Local Accounts



Port scanning for discovery has three configurations or controls:

- Port Scan Enable: Whether to port scan at all. Defaults to unchecked.
- Port Scan Timeout: How long (in seconds) the port scan will try before giving up. Defaults to 30.
- Port Scan List: A comma-delimited list of ports to scan. These depend on the configuration of the systems you will scan. Defaults to NetBIOS (135) and Active Directory services (445).

Examples of scanners that have a port-scanning timeout option for Active Directory include:

- Windows local accounts
- Active Directory user accounts
- All dependency scanners

Accessing Port Scanning

Simply go to **Admin > Discovery Configuration > Edit Discovery Sources (button) > Configure Discovery Scanners (button) > Accounts (tab)**, and then click the pencil icon for the desired scanner. If the configurations are on that page, that scanner supports port scanning. See the previous figure.

Additional Reasons to Consider Discovery Port Scanning

Lowering the Discovery Scanner Timeout May Cause Issues

If you lower the regular discovery scanner timeout, without port scanning enabled, you may kill a running scan. In addition, non-Active-Directory discovery scanners, such as a custom PowerShell scanner, that are slow or prone to hanging may also be disrupted or even crash if the regular discovery scanner timeout is set too low. As a best practice, we recommend enabling port scanning and not lowering the regular scanner timeout, which defaults to five minutes, unless Delinea Support asks you to. Do not lower the port scanning timeout below 15 seconds.

Secrets with Multiple Dependencies May Create Especially Long Timeouts

Without discovery port scanning enabled, discovery scanners rely on the standard timeout, which defaults to five minutes. If a secret has multiple dependencies, the system may have a chain of discovery timeouts to process, one at a time. With the default five-minute timeout on all the systems, timing out can take a long time, especially if you have a lot of machines turned off or unavailable. Discovery port scanning greatly reduces that.

To calculate the maximum timeout for discovery use this formula (with all systems using the same timeout value and each secret having the same number of dependencies):

```
(number of secrets) × (number of dependencies) × (timeout value) = (maximum minutes for discovery scans)
```

For example, using the default five-minute timeout value for 35 secrets, each with three dependencies:

```
35 × 3 × 5 = 525
```

Thus, 8.75 hours (525 ÷ 60) of timeout are possible and enabling discovery port scanning becomes a really good idea, especially if you have a lot of machines down at any given time.

> We can ignore clustered objects as part of a discovery scan, but we cannot ignore disabled computer objects, so Secret Server tries to scan each object that exists within AD. If you have a centralized area for disabled computer objects, consider configuring discovery to be OU specific and excluding your disabled computers OU to make discovery more efficient.

- Windows enforces a maximum time limit for a response to TCP Syn.
- The first attempt runs 3 seconds, then it retries with increasingly long limits.
- The number of retries is determined by MaxSynRetransmissions which can have a value of 2-8.
- MaxSynRetransmissions Maximum Time Windows will wait:
  - 2: 7 sec
  - 3: 15 sec
  - 4: 21 sec

  - 5: 63 sec
  - 6: 123 sec
  - 7: 183 sec
  - 8: 243 sec
- To prevent timeouts, the customer should update to Secret Server 11.6 or greater and execute "netsh interface tcp set global MaxSynRetransmissions=N" on the Windows server the Delinea Distributed Engine is executing on.
- Choose a value for N that corresponds to a Windows timeout greater than the Discovery Port Scan Timeout.
- Example: a Discovery Pre Scan Timeout of 30 seconds, MaxSynRetransmissions should be set to MaxSynretransmissions=5 which will cause the Windows TCP stack to wait up to 63 seconds (which is the lowest value which is greater than or equal to 30 seconds).

When to Run Discovery

Currently, you cannot set when discovery runs via a control or setting. You can, however, approximately set when it runs by disabling and enabling it at the desired time. It runs daily around the same time as when it was first enabled and then again according to whatever the discovery scan offset hours interval was set to. If you are running discovery once per day, we suggest:

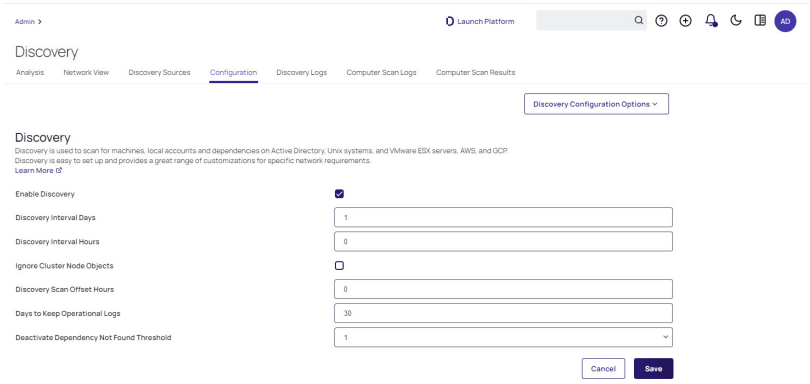
- Choosing a start time outside your normal business hours, such as midnight.
- First running several ad-hoc discoveries when your network traffic normally drops at the end of the day. Record how long each discovery process takes. Remember, this can vary greatly if a lot of machines are down, which is why we suggest conducting more than one discovery.

  Running a test with discovery port scanning disabled may provide valuable insights into the differences in performance or results.
- Using the average time the test runs took, calculate when to start discovery at a time when no anticipated portion of the discovery period is during your high-traffic times. We suggest having an end buffer as long as possible to account for variability, so if your average discovery time is fairly long, it might be best to start discovery soon after your network traffic drops off for the evening. This is especially true if your machine pool is growing.

For example, if your tested average discovery time was four hours and your network traffic is busy between 0600 and 1800, you should run discovery between 1800 and 0200, the closer to 1800 the better.

Discovery Settings

Figure: Discovery Settings Page



The settings are:

- Discovery interval for days and hours: How often you want the regular discover scan to occur.
- Ignore Cluster Node Objects: Tells Secret Server to not run discovery on machines identified as "msclustervirtualserver." Do not change this setting.
- See Discovery Scan Offset Hours for a discussion of the last setting.
- Deactivate Dependency Not Found Threshold: If set to 0, this setting means we will never disable dependencies if they are not found during a scan. If set to a higher number, it would indicate the number of failed attempts to find the dependency before finally disabling it". The threshold can be set to *Never* to prevent disabling, or a count for times a dependency is not found.

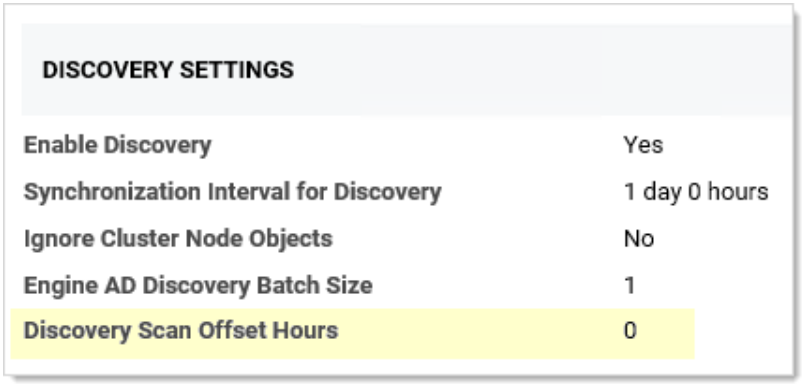
> There is another "Discovery Batch Size" setting on the Advance Settings page, which is usually only available to Delinea Customer Support. This setting, too, is legacy, and should not be set.

Environment-Specific Considerations

Discovery Scan Offset Hours

This section discusses a setting that allows you to quickly discover changes without greatly increasing traffic.

Figure: Discovery Settings Page in View Mode



The "discovery scan offset hours" (DSOH) setting is for customers that need to detect new (to the network) systems quickly without excessive network traffic during business hours. For example, you might need this feature if you have lots of server testing (systems are up and down) or laptops (systems are connected or not). The trick is doing this while minimizing the networking load.

We accomplish this with discovery scan offsets. With these, you have multiple synchronization scans per day, rather than just one, where Secret Server attempts to scan each and every system, but first Secret Server looks up each system to see if that system is flagged for scanning. The process goes like this:

1. Initially, Secret Server scans each discovered system and resets its DSOH timer, which is set to the number of hours defined by the DSOH setting value. Secret Server has a separate timer for each scanned system.
2. Once set, each timer starts counting down. Until that timer runs out, Secret Server ignores the scanned system if it runs a discovery scan.
3. When the timer is finished, the system is again flagged for scanning.
4. The next time Secret Server does a discovery scan, it sees the flag is present and scans the system.

The period the "scan me" flag is down (the period the timer is running) is defined by the DSOH setting. Thus, DSOH essentially tells Secret Server how long before scanning that discovered system again.

For example, if you have a discovery scan offset of 12 hours and a discovery interval of four hours:

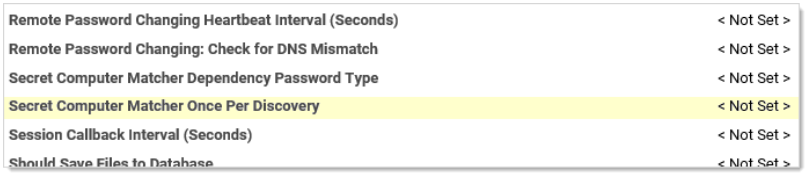
1. Start: The first time discovery runs, it scans every object because each one's timer is zeroed out, which makes it flagged for scanning. After scanning, each object's timer starts to count down, which makes it unflagged for scanning.
2. At four-hours: The next time discovery runs , it ignores the objects that were scanned the first time (because their timer was set to 12 hours), but it does process any newly discovered objects.
3. At eight-hours: In four more hours the same happens—only new objects are processed.
4. At 12 Hours: In four more hours, the scan runs again. This time, the 12-hour scan offset has expired, and all the timers of the original objects are zeroed out. The process begins anew—discovery scans every object because its timer is zeroed out, which makes it flagged for scanning. After scanning, each object's timer starts to count down, which makes it unflagged for scanning.

Advanced Settings

These settings reside in the ConfigurationAdvanced.aspx file, which you should not edit unless Delinea Support asks you to.

Run Secret Computer Matcher Once per Discovery

Figure: Secret Computer Matcher Once per Day

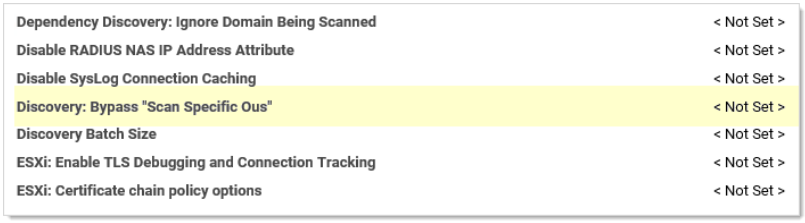


During the discovery process, secrets are matched with their machine. For smaller customers, this likely has little performance impact. For very large customers, the performance impact is noteworthy. We recommend that large businesses enable this option to decrease matcher resource use.

By default, the secret computer matcher runs once every five hours (this is non-configurable). This means the matcher runs four times per day, and only one of those times could coincide with discovery running at four-hour intervals. The other three will not run in tandem with discovery and thus will increase network traffic. If you enable this setting, the matcher will instead run after each discovery completes. If discovery only runs once, the matcher only runs once too. This more efficient because discovery can take hours to run, and having the matcher run several times during that period wastes processing.

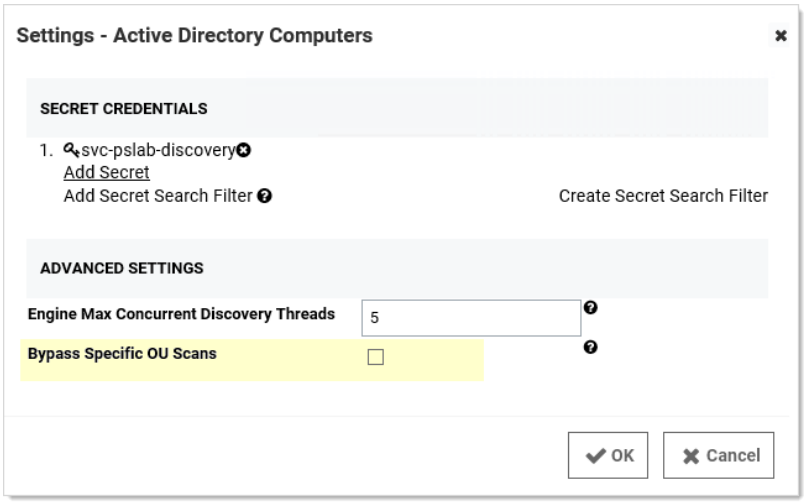
Limit the Network Traffic Caused by Nested Organizational Units

Figure: Discovery: Bypass "Scan Specific OUs"



If you configure discovery for Active Directory to scan by separate OUs and not by the entire domain, nested OUs can overwhelm your message bus. This occurs because each OU generates its own message unless you enable this setting. So if your enterprise has a complex tree of nested OUs, as many large businesses do, you could experience this issue. Smaller enterprises with single or a small number of nested OUs can ignore it. If you change the configuration in the Advance Configuration page file, it will affect all discovery source settings (some scanners have a similar configuration that only affects them). Alternatively, for more flexibility, you can configure this individually at the scanner level by checking the Bypass Specific OU Scan check box on the Settings - Active Directory tab for the scanner:

Figure: Tuning Active Directory Settings



Engines and Engine Workers

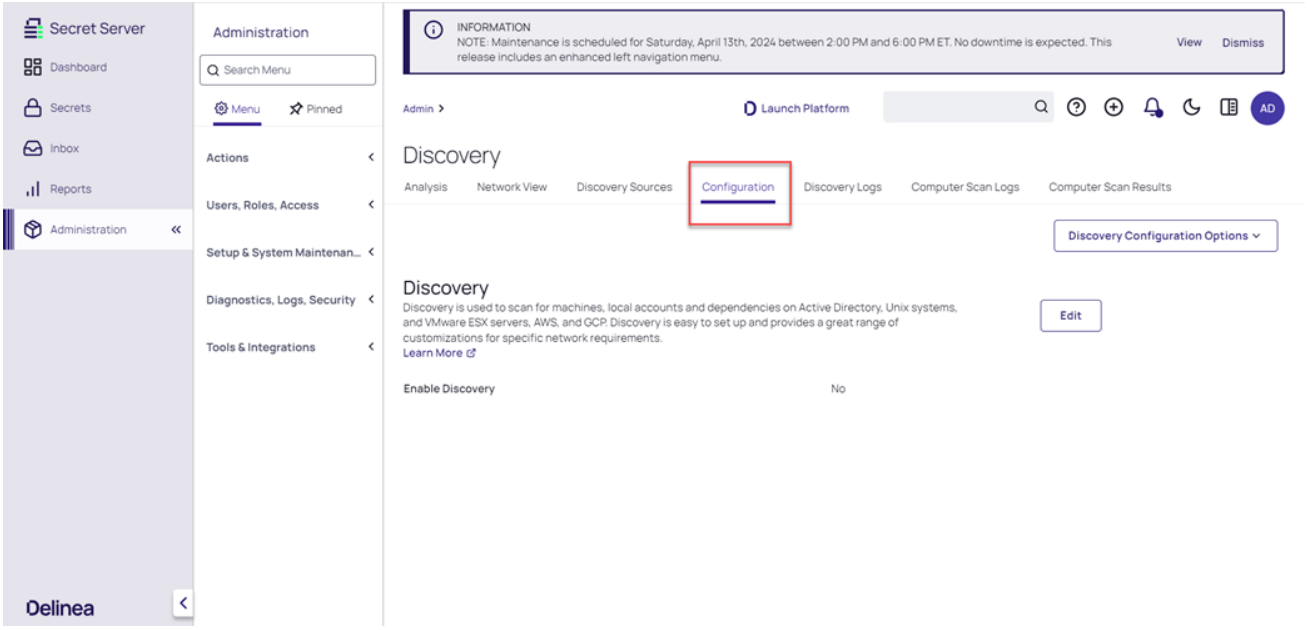
The number of distributed engines and engine workers within your environment can affect how fast discovery completes. Increasing CPU counts on your existing engines may help them to complete a diverse set of tasks more efficiently but might not have much effect on discovery processing time. If an engine is doing discovery, only a subset of consumers run and they will run into a prefetch count limit (30 messages per engine). Thus, increasing the number of engines and engine workers might decrease total discovery time by increasing that prefetch limit.

Domain Name Index

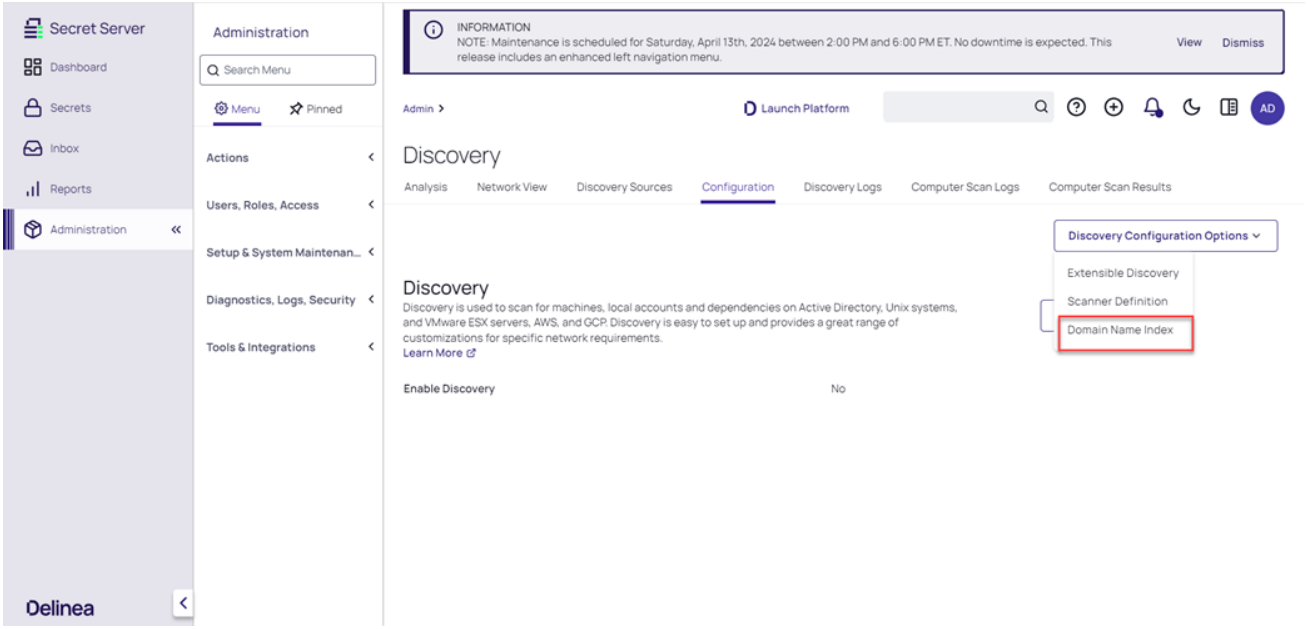
During Discovery, Secret Server needs to associate discovered accounts with domains and secrets. Discovery may return different forms of the domain name, so doing various forms of lookup to try and get the Discovery domain is frequently required. The Domain Name Index page exposes this list of mapped names to the admin and allows the admin to add their own known mappings. This helps to reduce the number of lookups required and therefore improves Discovery's speed and accuracy.

To access the Domain Name Index functionality:

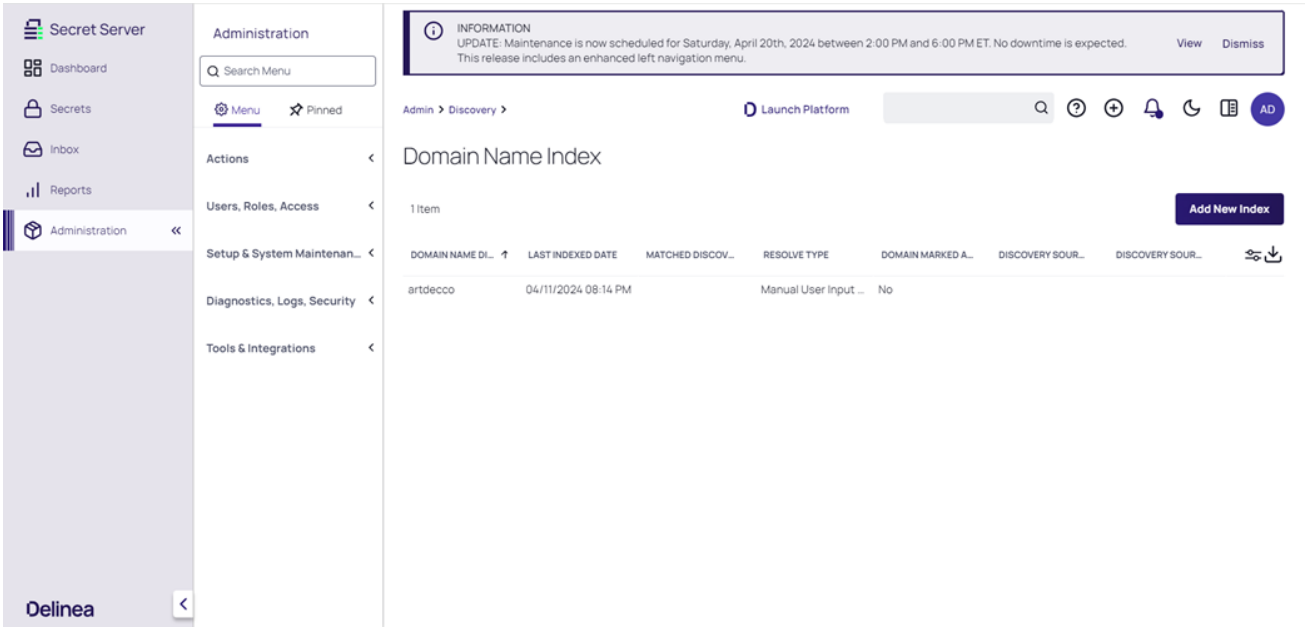
1. Navigate to **Administration>Discovery.**
2. Click the Configuration tab.



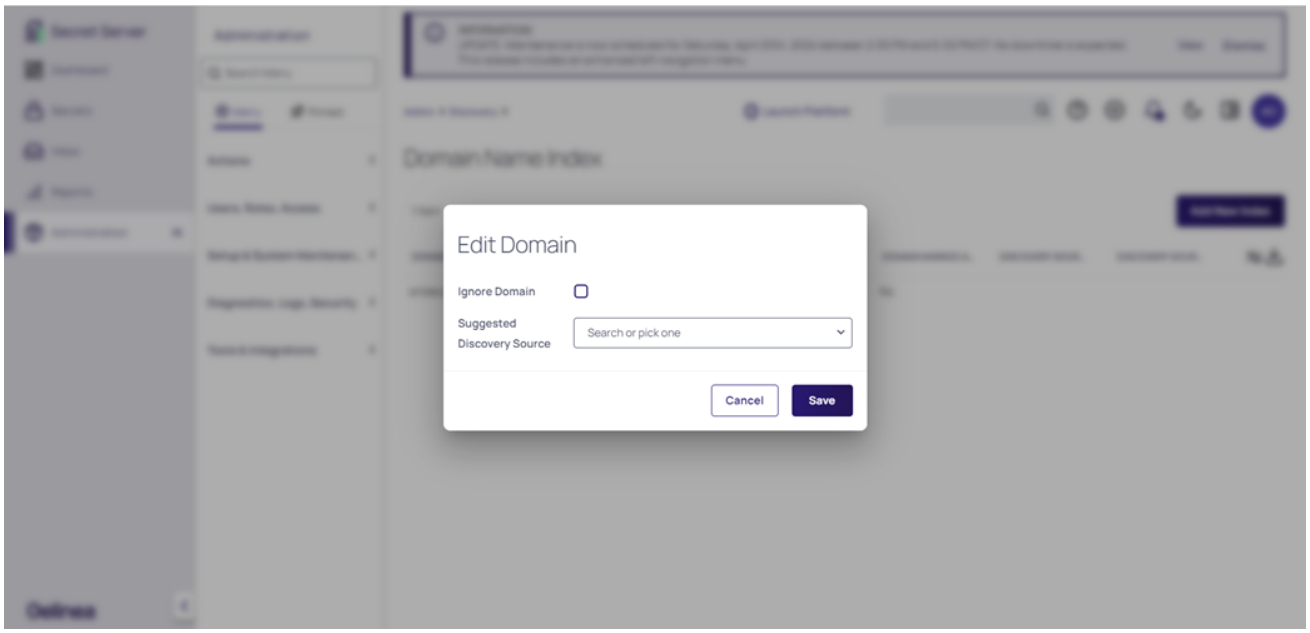
3. Open the *Discovery Configurations Options* menu and click **Domain Name Index**.



4. A list of list of mapped names to the admin will be displayed



5. **Optional:** Admins can add their own known mappings



Discovery Error Messages

The following are common error messages received when performing discovery and their possible causes:

User credentials cannot be used for local connections

This error typically occurs when attempting to run discovery on the server that Secret Server is running on, due to WMI restrictions.

No AD Account Services

No services run by Active Directory accounts have been found on the machine.

Computer is inaccessible or does not exist

Port 135 is blocked.

The target computer could not be reached

The machine is not connected to the network.

Access Denied

The account used to sync the domain with Secret Server does not have domain admin or local admin privileges for the machine it is attempting to scan for accounts.

Bad parameters - Script Error: Cannot bind argument to parameter 'Message' because it is null.

There is a mismatch between parameters referenced by the script and the arguments passed in. Check the script arguments on the scanner or dependency changer against the script.

Discovery Network View

Overview

The discovery network view allows users to view the results of completed discovery after defining the discovery sources. This view is crucial for managing the discovered items and provides information about each item.

> Items from disabled discovery sources will not be visible.

To filter items, users can select the type: computers, computer accounts, public keys, service accounts, or directory accounts. The network view tree and advanced filters can be used to view the organizational structure.

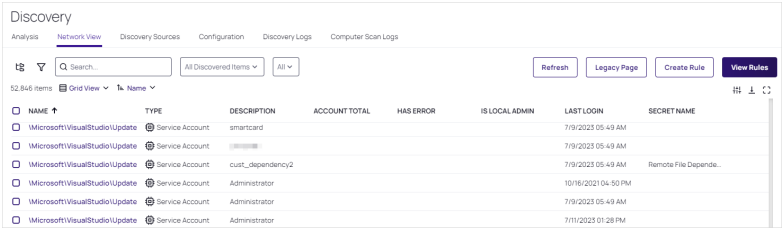
By selecting any row, users can open a preview pane that displays a summary of the item. The preview panel also allows expanding to see items discovered on that specific item. For example, expanding a computer will reveal all local computer accounts. Users can initiate a rescan of certain items like computers by clicking the scan button. Additionally, a link is provided to access a full details view of the computer.

Viewing the details of an item provides more information about related items. In the case of a computer, there is an option to connect to the computer if it is managed or has an associated secret.

All items are interlinked with their respective details. For instance, while viewing a computer, users can access information about all the services and check if any service is running as a directory account. The directory account is also linked, allowing users to view all computers with services running under that directory account.

Procedures

1. Click the **Administration** button on the main menu. The All Settings page appears.
2. Click the **Discovery** link in the **Core Actions** section. The Discovery page appears on the Discovery Sources tab.
3. Click the **Network View** tab. A graphical summary of your discovery activity appears:



4. **Filtering by network view source:** By default, all types or sources are displayed. To filter that, click the folder icon on the top-left of the page. The Network View panel appears. Drill down in the tree to choose what source you want to limit the view to.

   If you close the panel, the view stays in effect–you must click **All Sources** to clear the view.

5. **Filters panel:** Click the funnel icon at the top-left of the page. The Filters panel appears. You can filter by:
   - **Item type:** Click to select one of the **Item Type** selection buttons: computers, computer accounts, public keys, service accounts, or directory accounts.
   - **Management status:** Click to select one of the **Managed** selection buttons.
   - **Created date:** Click the left side of the calendar box to enter or select the start date. Click the right side of the calendar box to enter or select the end date.
   - **Last polled date:** Click the left side of the calendar box to enter or select the start date. Click the right side of the calendar box to enter or select the end date.
   - **Lasted reached date:** Click the left side of the calendar box to enter or select the start date. Click the right side of the calendar box to enter or select the end date.
   - **Scan Template:** Click to select one of the scan template selection buttons. Scan templates are secret templates that define the criteria needed for discovering a given type of asset, such as an account type or host range.
6. **Grid or card view:** Click the **Card/Grid View** dropdown to toggle between views.
7. **Sorting:** Click the sorting dropdown (on the left) to choose what to sort the list or cards by. Click the same choice again to toggle between ascending and descending. Alternatively, in grid view, you can click the column heading to do the same thing. The dropdown provides sorting in card view and allows you to sort on hidden columns.
8. **Hiding and ordering displayed columns:** Click the displayed columns icon on the top far right of the page. The Displayed Columns panel appears. Click any of the checkboxes to hide or display a column. Drag the icon on the left to change the display order of the columns. Click the **Save** button when you are done.
9. **Creating Rules:** As mentioned above, rules define criteria that run against already discovered accounts and dependencies. As newly discovered accounts are found, rules can automatically take them over, manage them, or send alert notifications. To create a rule:
   a. Click the **Create Rule** button. A Create Rule dialog box appears.
   b. Click the **Rule Type** dropdown list to select a type:
      - Accounts: Automatically create secrets or send emails when local accounts or public keys that match the rule criteria are discovered.
      - Dependencies: Automatically add a dependency to existing secrets. No Secrets are created.
   c. (optional) Type a text string that appears inside the computer's name in the **Computer Name Contains** text box (account rules only).

d. (optional) Type a text string that appears inside the account's name in the **Account Name Contains** text box (account rules only).

e. (optional) Type a text string that appears inside the operating system's name in the **Operating System Contains** text box (account rules only).

f. Click to select the **Manage Accounts** check box to indicate that secrets will be created and the service and accounts will be managed by the vault.

g. Click the **Create Rule** button.

10. **Viewing existing rules:** Click the **View Rules** button.

Discovery on Non-Domain Joined or Unix Targets

Overview

When running Discovery on non-domain joined targets or Unix targets, there are two methods of finding local administrator credentials to authenticate to the target:

> The two methods can be used together.

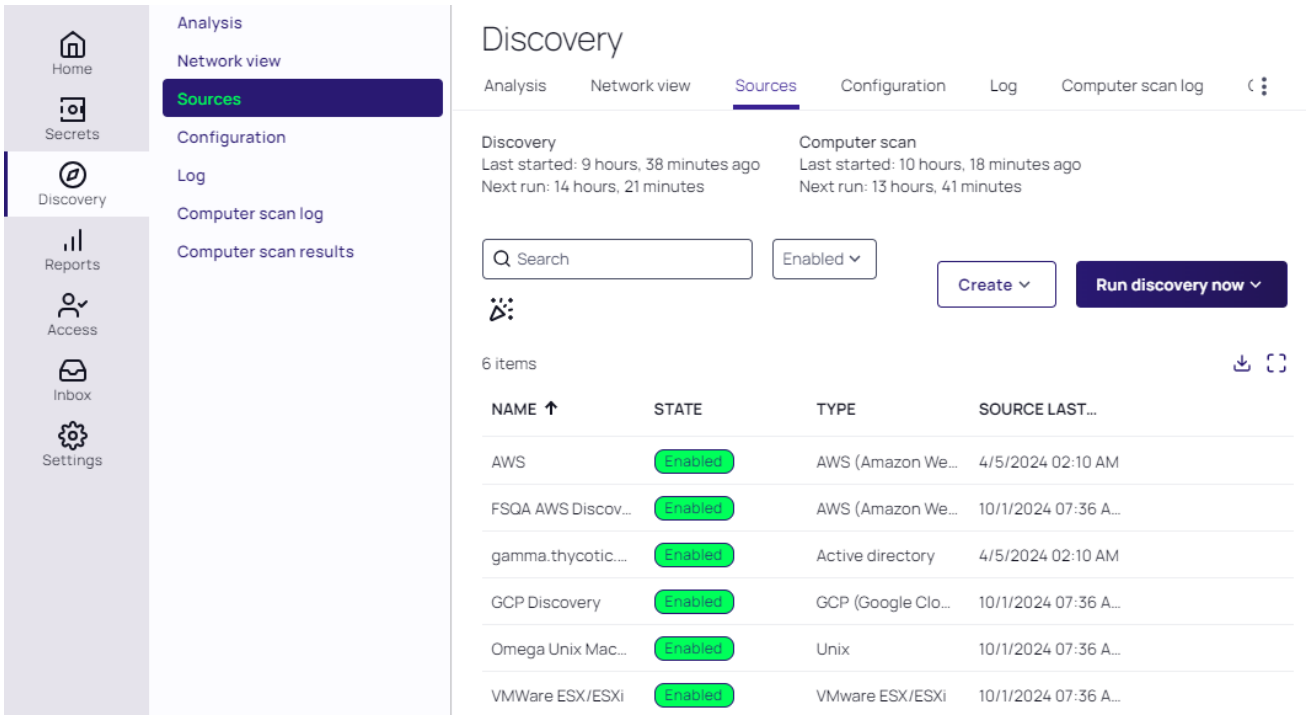
- **Specify a secret with an expected default password** - recommended for performing an initial scan if you have a known password or key for a privileged account.
- **Specify a Secret Search Filter** - recommended when you cannot use a default password because each machine's account password is unique.
  A **Secret Search Filter** dynamically searches for a secret with a name or folder location that corresponds to the target scanned. If a matching secret is found, Secret Server will authenticate to the target using the administrator credentials in the secret.

> To use a secret search filter, the administrator account names must exist as secrets in Secret Server and they must follow a regular naming pattern.

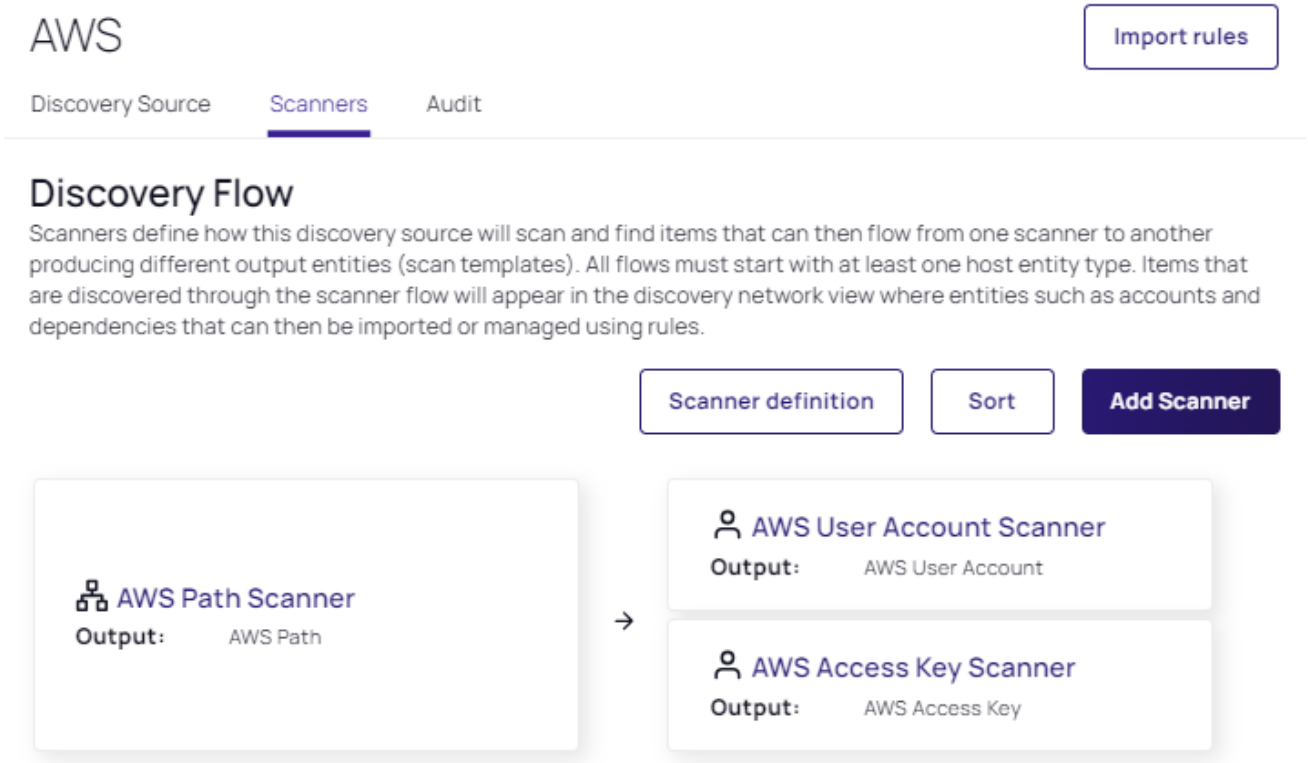
> The discovery secret search filter is available in Secret Server 10.0.000006 and newer.

Setting Credentials on a Discovery Scanner

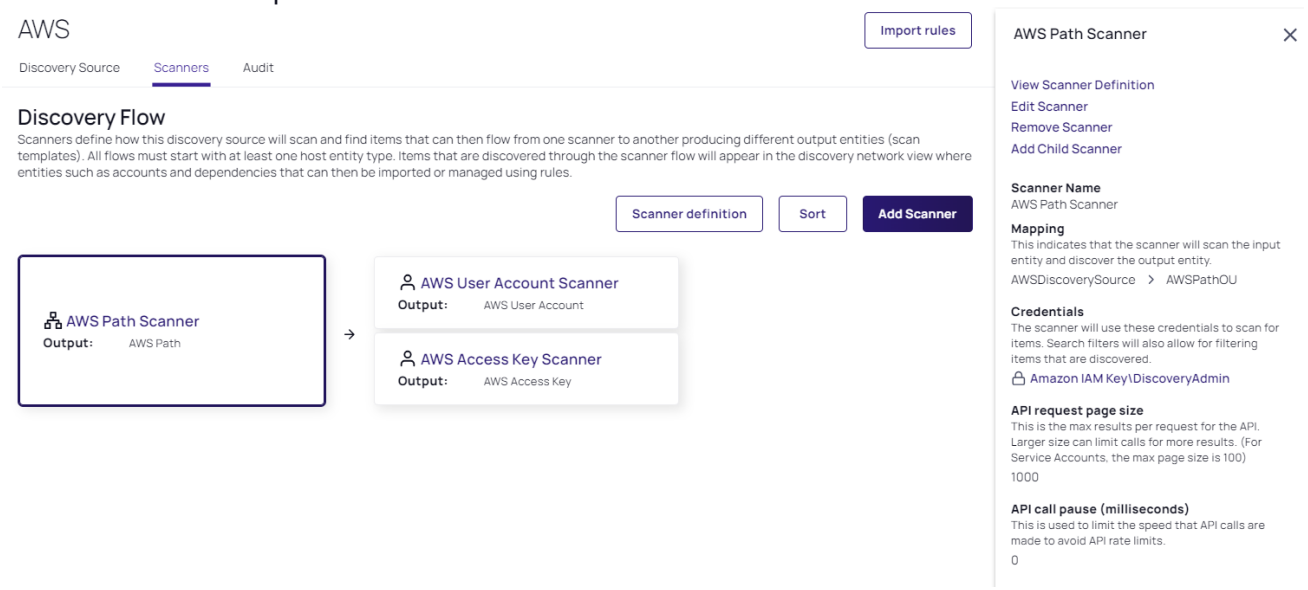
1. From the left menu-bar, hover over **Discovery** and select **Sources**, the page will open by default on the **Sources** tab:



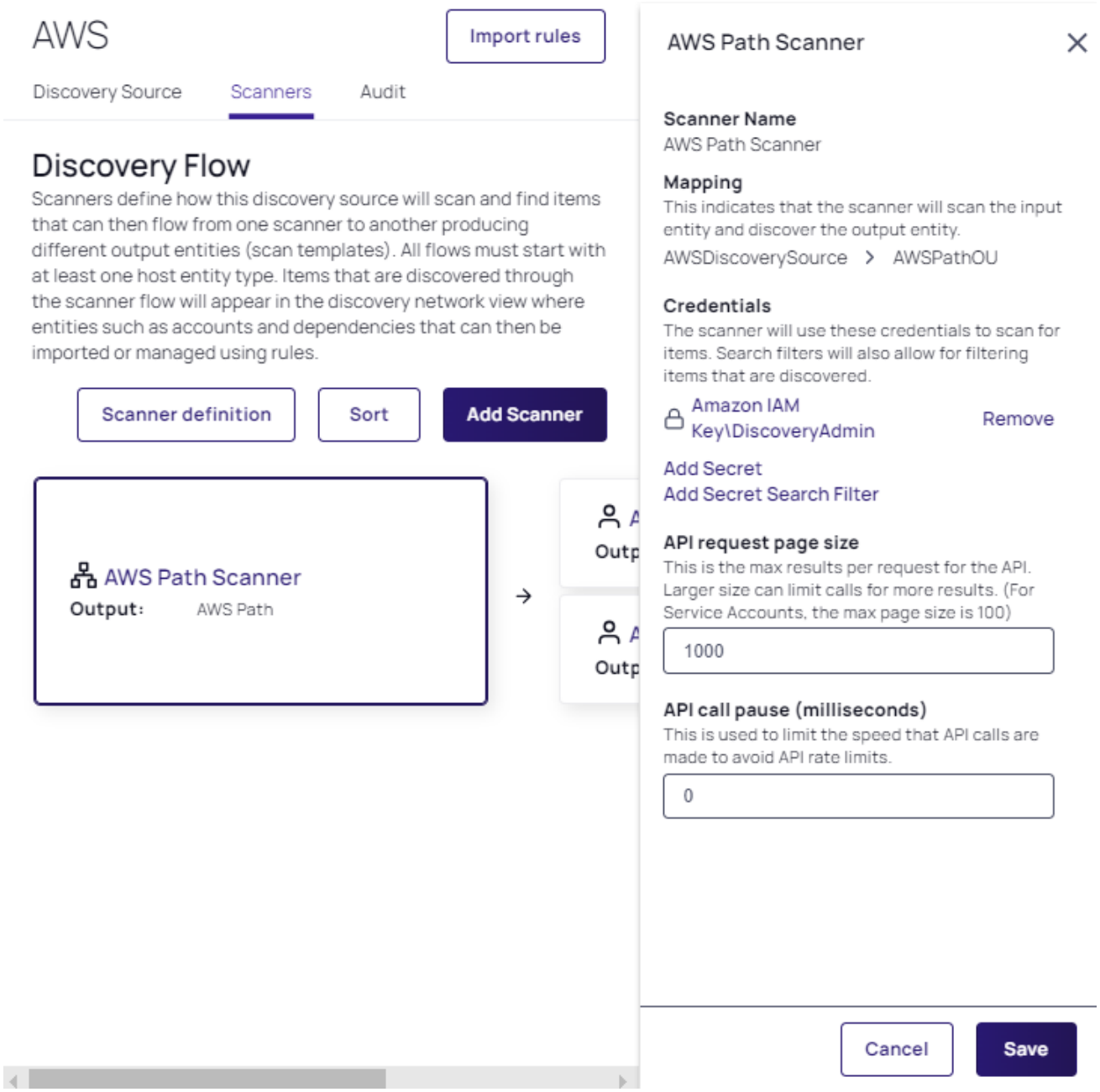
2. Select one of the enabled discovery sources as shown above, and inside that page, click on the **Scanners** tab:



3. Select a scanner from the ones available, a details page will popup on the right side of the screen where you will see the **Edit Scanner** option:



4. Select the **Edit Scanner** option and here you will see different settings for that scanner. Under **Credentials**, choose among the following options:
   - Click **Add Secret** to specify a default credential.
   - Click **Add Secret Search Filter** to specify an existing secret search filter.

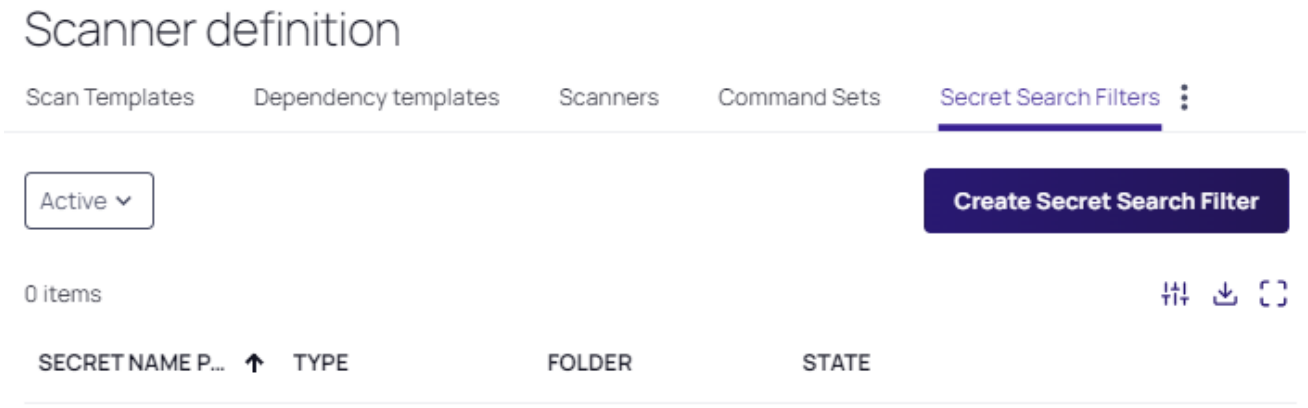


Secret Server will try the secrets and secret search filters in sequence until it finds a match.

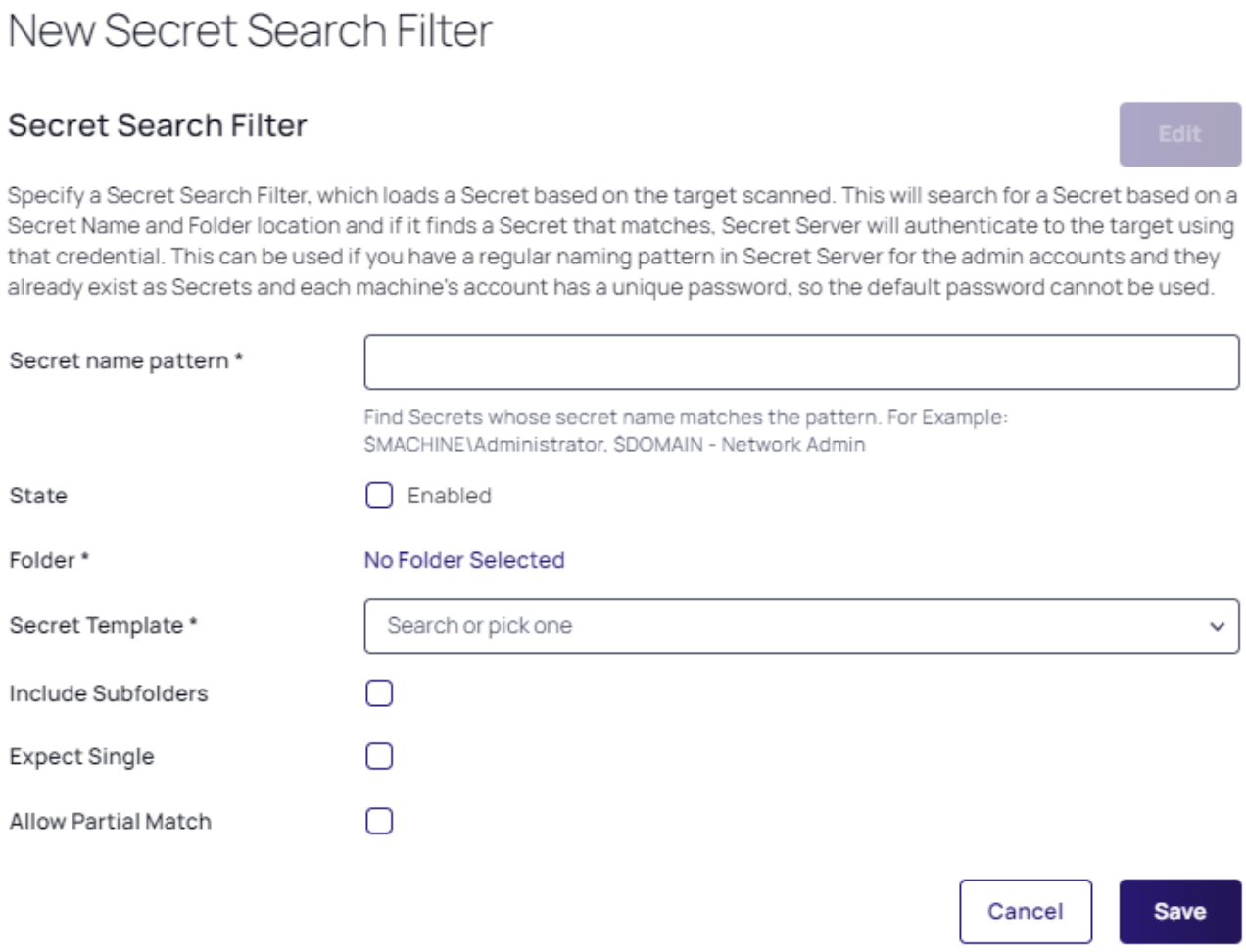
Creating a Secret Search Filter

If you decided to create a secret search filter, perform the following:

1. In the search bar type **Discovery secret search filters**, this will cause the **Scanner definition** page to appear:



2. Select **Create Secret Search Filter** and the following options will appear:



3. Specify all the mandatory* settings along with any others you need, as described below:

- **Secret Name Pattern**: Specifies the pattern that Secret Server will search for. The search is dynamic based on the target. For example, if scanning a machine named `appserver01`, Secret Server will also search for a secret named `appserver01\system`.
- **State**: Enabled or not.
- **Folder**: Specifies the folder to search within.
- **Secret Template**: Specifies the template that returned secrets should be based on.
- **Include Subfolders**: Specifies that the search should include the specified folder as well as subfolders.
- **Expect Single**: Specifies that only one result should be returned. If more than one is returned, Secret Server will log an error to the discovery log.
- **Allow Partial Match**: Specifies that secret names will be returned if they partially match the pattern. By default the secret name must be an exact match to the secret name pattern.

4. Click **Save** when all your settings have been speciffor exampled and go back to your discovery source, ie. back to the **Edit Scanner** page.
5. Click **Add Secret Search Filter** and select the filter you just created.

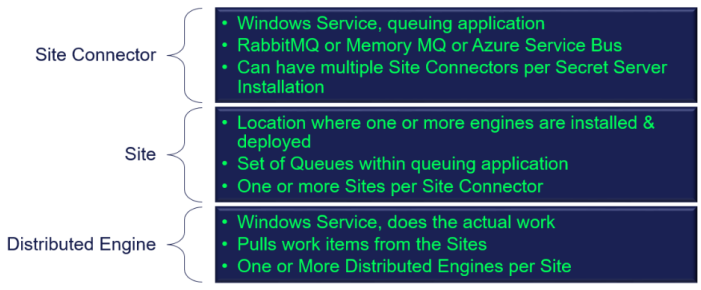
Now, when scanning a machine, Secret Server will try a default credential, and then it will try the secret returned by the search filter.

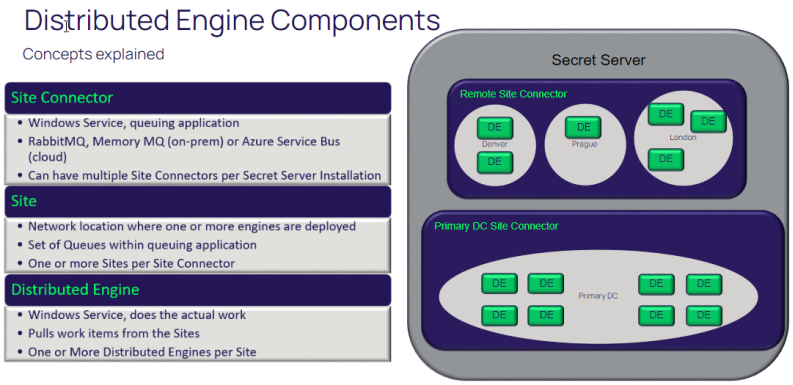
Discovery and Sites–Where Does Secret Server Run Discovery Scans?

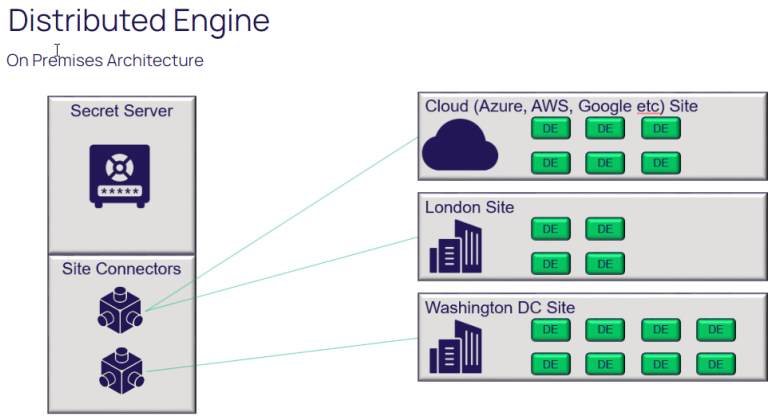
Like many operations in Secret Server, you can configure discovery to run locally on IIS machines running Secret Server using website processing or by running through a distributed engine. Distributed engines are agents that you can deploy to remotely process work. They are useful for scenarios where performance is an issue or the work must take place in a remote network where the ports required by discovery are not available. You can configure discovery to use a single site location per discovery source or on a per-OU basis for AD.

- A site refers to a collection of Distributed Engines. Secret Server interacts with the site, allowing any available Distributed Engine within the pool to execute tasks as needed.
- A site also denotes the network of queues utilized by a pool of Distributed Engines to facilitate communication with Secret Server. This setup ensures efficient task distribution and management across the engines.

Distributed Engine Components

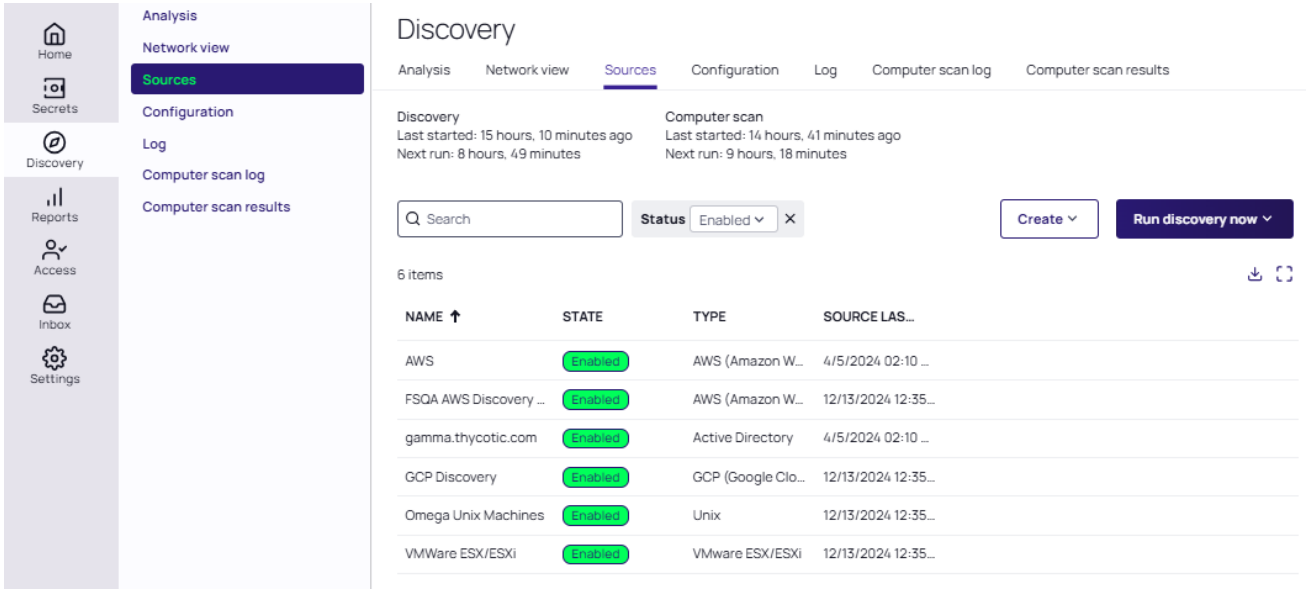




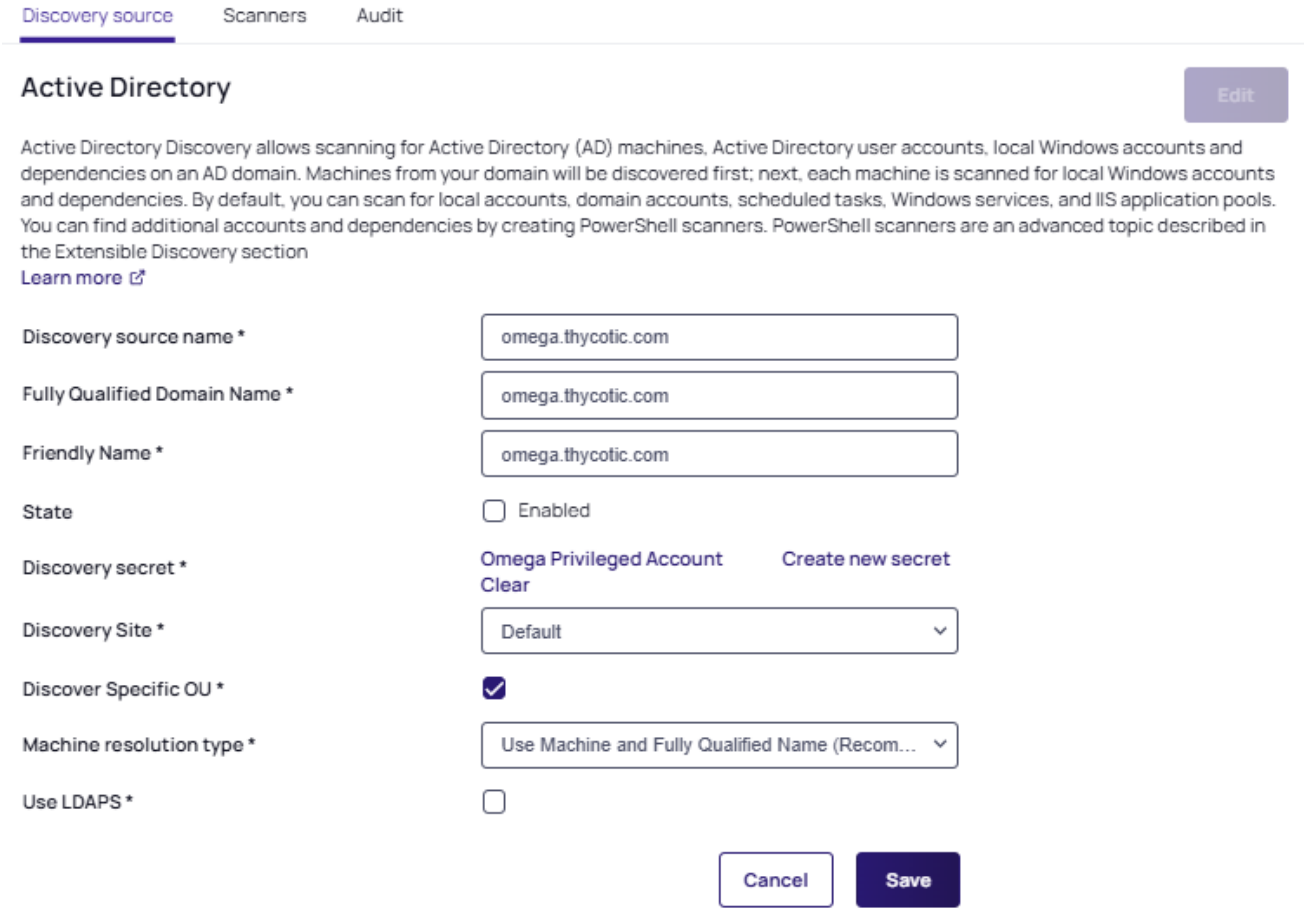


Enabling Specific OU Domain Discovery

1. In the left-hand menu select **Discovery > Sources**. The Discovery **Sources** tab appears:



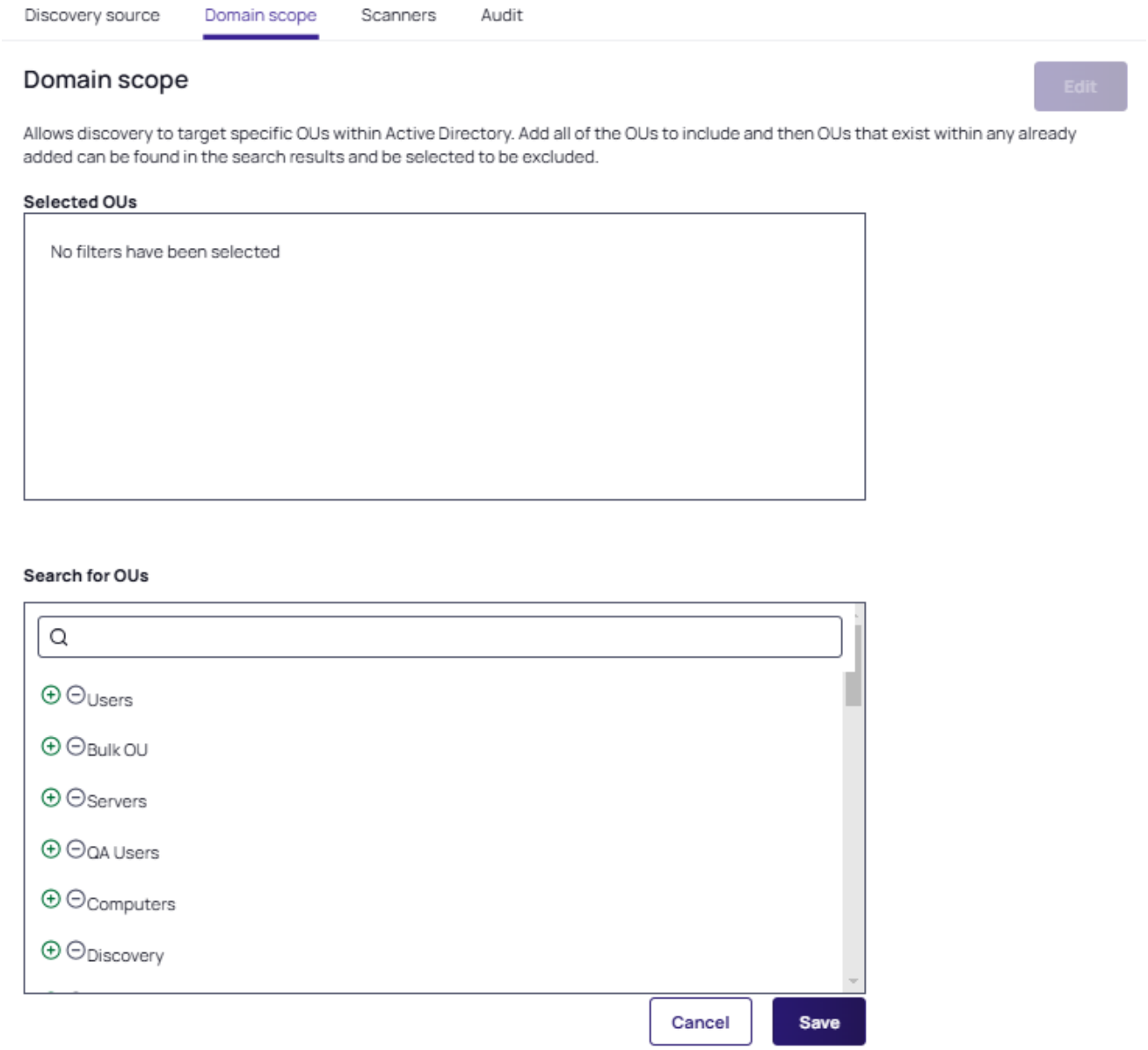
2. Select one of the discovery sources listed which you want to configure. The **Discovery Source** page and tab for that source appears, with all its details. These include a brief description, the Discovery source name, Fully Qualified Domain Name, Friendly Name, State, Discovery secret type, Discovery Site, Discover Specific OU, Machine resolution type, and if Use LDAPS has been selected.
3. Click the **Edit** link next to the discovery source name. The page becomes editable.
4. Click to select the **Discover Specific OU** check box:



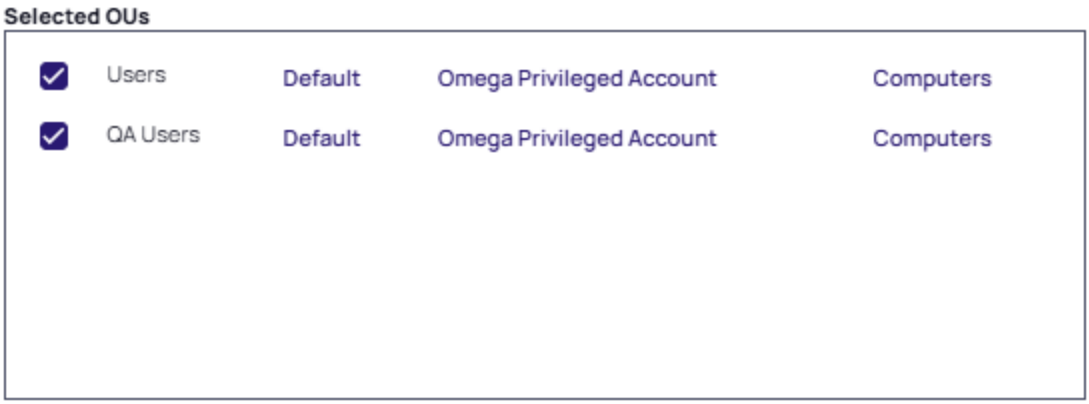
When **Discover Specific OU** is enabled, unmapped secrets are not added.

5. Click the **Save** button. The **Domain Scope** tab appears.
6. Select the **Domain Scope** tab.

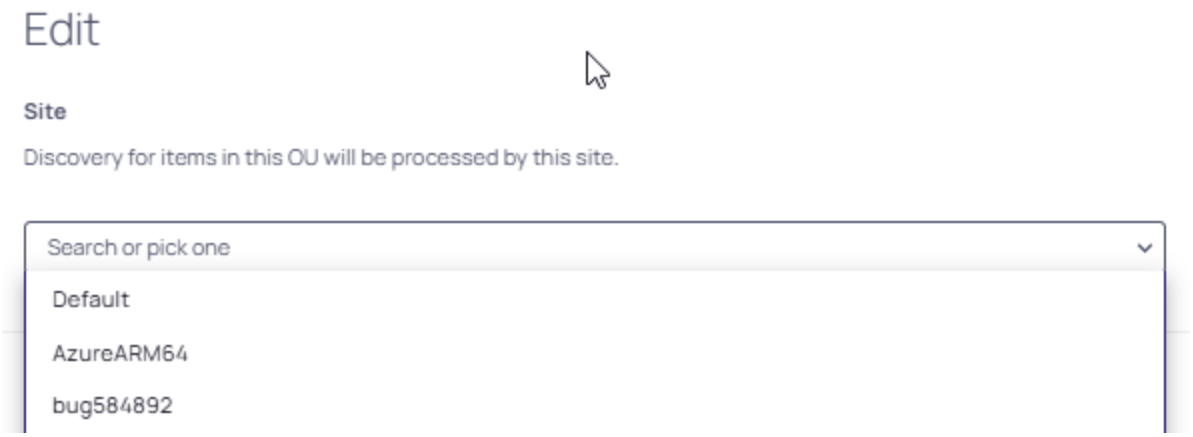
7. Click the **Edit** link next to **Domain Scope**. The page becomes editable:



8. In **Search for OUs** type the name of the desired OUs. Matching OUs appear.
9. Click the circled plus button to add an OU. Upon addition, OUs appear automatically in the **Selected OUs** box above the search list. Clicking the checkbox next to Selected OUs removes them:



10. Click the site link (e.g. Default, shown above) to edit the site (distributed engines) included. A popup appears with a dropdown allowing you to choose what type of site will process the Discovery for items in this OU:



> Please be advised that if a privileged account is set under specific OUs, that the account will override any accounts set on the individual Discovery Scanners (Service, Scheduled Task, Local User, etc)

11. Click the credential secret link (e.g. Omega Privileged Account) to edit the secret that should be used when scanning items within this OU.
12. Click the scan target link (e.g. Computers) to edit what type of items should be discovered within this OU.
13. Repeat the previous steps for any additional OUs.
14. Click the **Save** button.

> The ports required for Discovery are documented in "Ports and IP Addresses Used by Secret Server" on page 765.

Extensible Discovery

> Extensible discovery is not covered by Delinea Support.

Overview

> It is important to have an understanding of built-in discovery scanning before attempting to create your own custom scanner. Please see "Introduction to Discovery Sources, Scanners, and Templates" on page 525 for more details.

Extensible discovery lets you extend the already powerful scanning abilities of Secret Server by creating custom scanners that run PowerShell. You can use either built-in or custom scanners and templates at each step of the extensible discovery process.

If the built-in discovery sources, scanners, or input and output templates, cannot meet your needs, you can use PowerShell scripts to perform any part of discovery. Doing so requires that you define your own input and output templates, as well as scanners, and then add them to a new or existing discovery source.

When to Use Extensible Discovery

Creating a discovery source using scripted scanners can be a lot of work to set up, so when should you consider it? If you only need to discover domain administrator accounts with standard dependencies (Windows services, application pools, and scheduled tasks), our built-in scanners will do the job, and extensible discovery is not necessary. However, your network probably contains other items you want to discover and bring under managed control. Here are some examples:

- Discover configuration files containing passwords and automatically add them as dependencies.
- Scan computers not joined to the domain.
- Create dependencies that run a SQL, SSH, or PowerShell script when a secret's password changes to log events to an external source, such as an external auditing or monitoring system.
- Record information not currently imported by local account discovery or custom fields in a secret template.
- Discover SQL Server logins as local accounts and import them as SQL Server account secrets.

> To run PowerShell scanners against machines for local account and dependency discovery, you may need to configure WinRM and CredSSP. See "Configuring WinRM for PowerShell" on page 1483 and "Configuring CredSSP for WinRM with PowerShell" on page 1479.

Extensible Discovery Tutorial

> Extensible discovery can be a long process for new users. It has many hands-on steps. As such, we believe a step-by-step tutorial is the best way to learn it.

Discovery scanners can run custom PowerShell scripts, as well as our built-in scanners for Active Directory, UNIX, and VMware ESXi. You can use one or more built-in or custom scanners at each step of the discovery process: host range, machine, local account, and dependency discovery.

Roughly, this tutorial has four tasks:

- Understanding the process
- Setting up scripts

- Creating scan templates
- Setting up discovery scanners and sources. You define dependency templates to change items manually added to secrets and added through discovery rules.

The tutorial shows you how to take full advantage of extensible discovery by creating an Active Directory discovery source that replaces each of the built-in scanners with a PowerShell script.

For simplicity, we will only create a Windows service dependency scanner in the dependencies step, but you can add additional built-in or custom PowerShell scanners to scan for application pools, scheduled tasks, or any other item that requires an action triggered when a secret's password is changed.

Task One: Understanding the Process

For most of this tutorial we will use the **Extensible Discovery Configuration** page as our launch page into each of the features that needs to be configured. Setting up extensible discovery requires making changes on several pages. The Extensible Discovery Configuration page has buttons linking to each of these pages as well as short explanations of what you need to do on them. The high-level process is as follows:

1. Create the scripts for each discovery step.
2. If necessary, create scan templates to define the information to return from the objects discovered at each step.

   > You want to avoid altering scan templates unless you absolutely need to. First, ensure the regular scanners cannot do the job. Once you change a template, you cannot use out-of-the-box scanners and must maintain your own PowerShell scripts for the local account and dependency scanners.

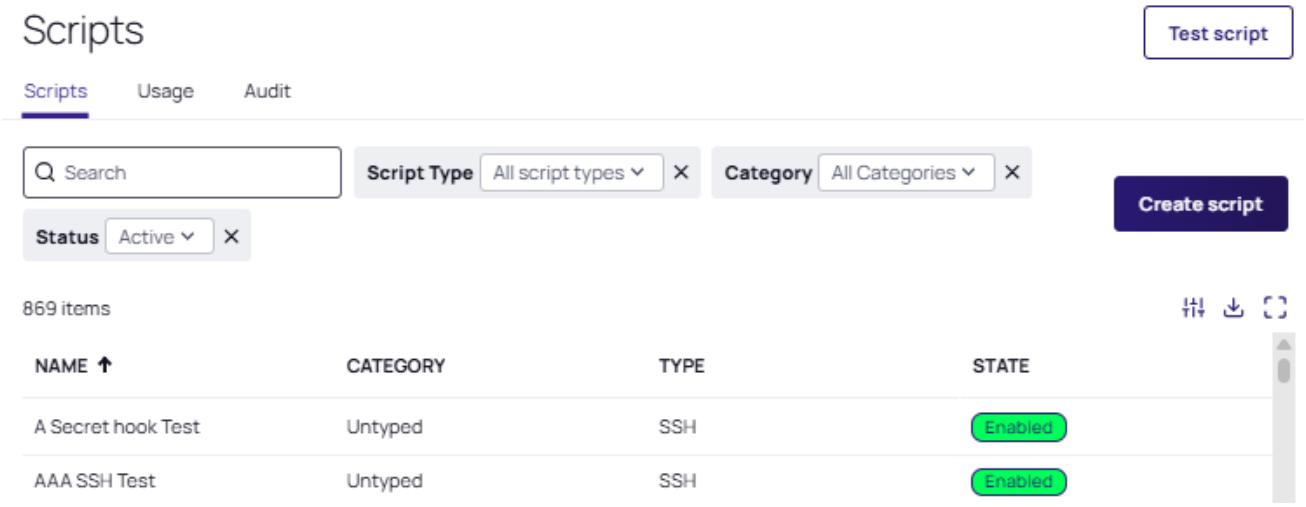
3. Create discovery scanners for each step to define:
   a. Which script to use for scanning.
   b. Which scan template represents the objects used as the source of data for the script.
   c. Which scan template represents the object returned from the script.
4. Create a discovery source that is configured to use discovery scanners in lieu of the default scanners.
5. Create a dependency changer for the type of dependency we want to manage.
6. Create a dependency template for the changer.
7. Manually add a dependency to a secret using the dependency changer.
8. Create a local account rule to import discovered accounts as secrets.
9. Create a dependency rule to import discovered dependencies as secrets.

Task Two: Creating the Scripts for Each Discovery Step

First, we add the scripts that we will use as our scanners to Secret Server:

> To use extensible discovery, you must use a Secret Server edition that supports scripts or has an "Advanced Scripting" add-on license.

1. Go to **Settings > All Settings > Tools and Integrations > Scripts: PowerShell, SQL, SSH**. The **Scripts** tab of the Scripts page appears:



2. Select the **Create script** button. The **New Script** page appears.
3. For the Host Range Scanner, Machine Scanner, Local Account Scanner, and Windows Service Dependency Scanner scripts, fill in the following fields:
   a. Use the script name in the **Name** text box.
   b. Fill in the **Description** text box with relevant details about the script.
   c. Select the **State** checkbox to make the script **Enabled**.
   d. Pick the **Script Type** from the dropdown list.
   e. In the **Category** dropdown list use the **Discovery Scanner** category for each script.
   f. Copy and paste each script listed below in code snippets into the **Script** code text box for each of the four scripts.
   g. Click the **OK** button and repeat the process for each script.

Include the following code snippets for each script mentioned above.

Script Name: Host Range Scanner

```
$passwordArg = $args[2]
               $username = $args[1]

               $domain = $args[0]

               write-debug "$domain $username $passwordArg"

               $distingisheddomain = "DC=" + ($domain.Split('.') -join ",DC=");

               $Spassword = ConvertTo-SecureString "$passwordArg" -AsPlainText -Force
#Secure PW
```

```
                $cred = New-Object System.Management.Automation.PSCredential
("$domain\$username", $Spassword) #Set credentials for PSCredential logon

                $ous = Get-ADOrganizationalUnit -filter 'Name -like "*"' -Server $domain
-Credential $cred | select-object -property Name, ObjectGUID, @{Name =
'DistinguishedName'; Expression = {$_.DistinguishedName.Replace
(",$distingisheddomain","")}}, ObjectClass

                return $ous

                # Script Args: $[1]$Domain $[1]$username $[1]$Password
```

Script Name: Machine Scanner

```
$Ou = $args[0];

                $domain = $args[1];

                $distinguisheddomain = "DC=" + ($domain.Split('.') -join ",DC=");

                if ($distinguisheddomain.ToLower() -eq $Ou.ToLower()) {

                $searchbase = $distinguisheddomain

                } else {

                $searchbase = "$Ou,$distinguisheddomain"

                }

                $FoundComputers = @()

                $ComputersinOU = Get-ADComputer -Filter 'Name -like "*"' -Server $domain
-SearchBase $searchbase -properties *

                foreach ($comp in $ComputersinOU) {

                $object = New-Object -TypeName PSObject

                $object | Add-Member -MemberType NoteProperty -Name ComputerName -Value
$Comp.Name

                $object | Add-Member -MemberType NoteProperty -Name DNSHostName -Value
$comp.DNSHostName

                $object | Add-Member -MemberType NoteProperty -Name ADGUID -Value
$comp.ObjectGuid

                $object | Add-Member -MemberType NoteProperty -Name OperatingSystem -Value
$comp.OperatingSystem
```

```
                $object | Add-Member -MemberType NoteProperty -Name DistinguishedName -
Value $comp.DistinguishedName.Replace(",$distinguisheddomain",'')

                $FoundComputers += $object

                }

                return $FoundComputers

                # args: $target $[1]$domain
```

Script Name: Local Account Scanner

```
$ComputerName = $args[0]

                $username = $args[1]

                $domain = $args[2]

                $password = $args[3]

                $objComputer = New-Object System.DirectoryServices.DirectoryEntry
("WinNT://$ComputerName", "$domain\$username" , $password)

                $children = $objComputer.Children | select-object SchemaClassName, Path,
Name, Properties, userflags, SIDType, Disabled

                $results = @()

                foreach ($child in $children){

                Write-Debug $child

                if ($child.SchemaClassName -eq 'User'){

                write-debug "adding to results"

                $object = New-Object -TypeName PSObject;

                $object | Add-Member -MemberType NoteProperty -Name Username -Value
$[child.Name](http://child.name/)[0];

                $object | Add-Member -MemberType NoteProperty -Name Resource -Value
$ComputerName;

                $object | Add-Member -MemberType NoteProperty -Name Disabled -Value
$child.Disabled;

                $results +=$object;

                }
```

```
                }

                return $results

                # Arguments $target $[1]$username $[1]$Domain $[1]$Password
```

Script Name: Windows Service Dependency Scanner

```
$ComputerName = $args[0]

                $accounts = Get-WMIObject Win32_Service -ComputerName $computername |
Where-Object{($_.StartName -like "*\*" -or $_.StartName -like "*@*") -and $_.StartName -
notlike "NT *"}

                if ($accounts) {

                $dependencyaccounts = @()

                foreach($dependency in $accounts)

                {

                $object = New-Object -TypeName PSObject;

                $object | Add-Member -MemberType NoteProperty -Name ServiceName -Value
$dependency.DisplayName;

                $object | Add-Member -MemberType NoteProperty -Name Enabled -Value
$dependency.Started;

                if ($dependency.startname.contains('@'))

                {

                $accountinfo = $dependency.startname.split('@')

                $username = $accountinfo[0]

                $domain = $accountinfo[1]

                }

                else

                {

                $accountinfo = $dependency.startname.split('\')

                $username = $accountinfo[1]
```

```
                $domain = $accountinfo[0]

                }

                $object | Add-Member -MemberType NoteProperty -Name Username -Value
$username;

                $object | Add-Member -MemberType NoteProperty -Name Machine -Value
$ComputerName;

                $object | Add-Member -MemberType NoteProperty -Name Domain -Value $domain;

                $object | Add-Member -MemberType NoteProperty -Name DependencyType -Value
'Powershell Script';

                $object | Add-Member -MemberType NoteProperty -Name AccountStatus -Value
'Expired';

                $dependencyaccounts += $object

                $object = $null

                }

                return $dependencyaccounts;

                }

                throw "Error - no service accounts found"

                return $null

                # args: $target
```

Task Three: Creating Scan Templates

The second task is to create scan templates for each object to be discovered. Scan templates define the types of objects that can be retrieved by discovery scanners. The goal of discovery scanning is to retrieve the following:

- Accounts that can be imported and managed as secrets.
- Entities (dependencies) requiring knowledge of password changes to managed secrets.

The process of finding these accounts and entities usually involves running the following scans beforehand:

- Host ranges where machines containing accounts can be found.

  For Active Directory, this usually involves scanning a domain for organization units or defining which organization units in a domain to check. For Unix and ESXi, this usually involves defining one or more lists of IP address ranges to scan.
- Machines to be scanned for accounts.

Each of these items—host ranges, machines, accounts, and dependencies—is defined by a scan template. The scan template specifies:

- At which step of the scanning process the item is created (scan type).
- The scan template from which the current template gets its required fields (Parent scan template).
- A list of fields that the item contains.

The **Fields** section describes the list of properties that will be returned by the built-in scanner or script for the item. At a minimum, you need to define one field for each of the fields on the parent scan template. For items that are returned from a script, you can define additional fields that the script will return on the object. These are mapped by name to the corresponding field on the scan template. These additional fields can then be used by future scanners.

Secret Server defines scan templates for all of its built-in scanners. Whenever possible, you should use these as input and output sources for your scripted scanners. Create your own scan templates if you need to capture additional information as data for your scripts or if you need to use specific input and output templates on the discovery scanners to drive multiple discovery workflows on a single discovery source. For this tutorial, we will create new scan templates for the output of each of our scripts.

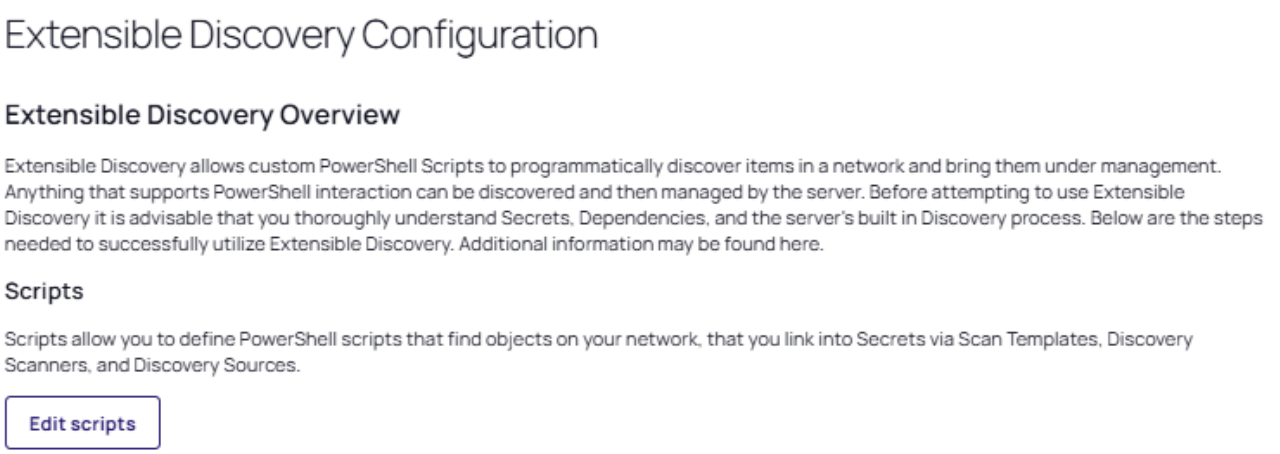
Host Range

The first scan template is the one that stores the results from our Host Range Scanner script. The script outputs an object with the following properties:

- Name
- ObjectGUID
- DistinguishedName

Our scan template must hence, have fields to store the values of these three properties:

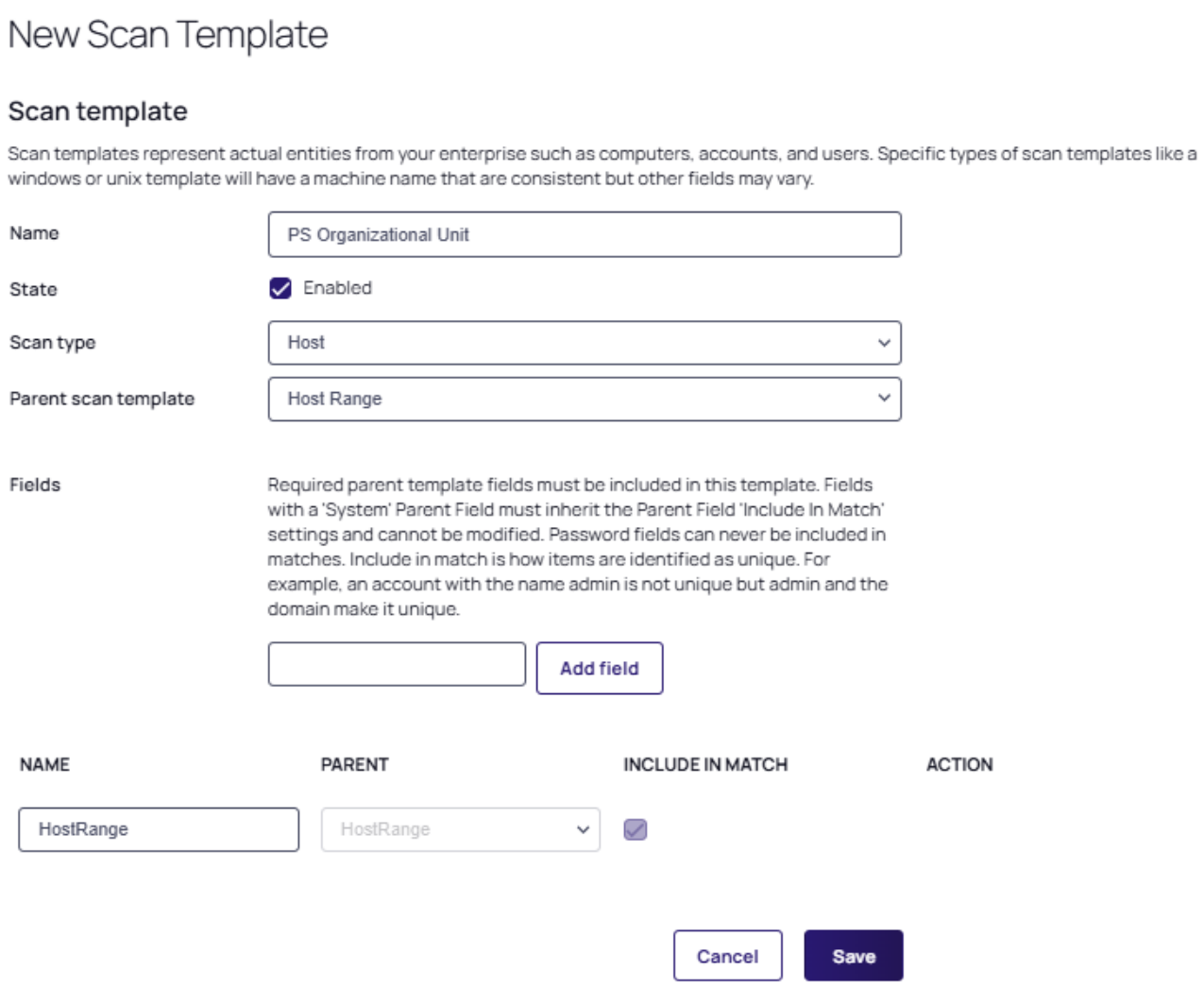
1. Go to **Discovery**.
2. Select the **Configuration** tab.
3. Click the **Discovery Configuration Options** dropdown and select **Extensible Discovery**. The Extensible Discovery Configuration page appears:



4. Click the **Configure Scan Templates** button. The Scanner Definition page appears on the **Scan Templates** tab.
5. Click the **Create Scan Template** button. The New Scan Template page appears.
6. Type `PS Organizational Unit` in the **Name** text box.
7. Select the **State** checkbox to make the template **Enabled**.
8. Leave the **Scan Type** set to **Host** in the dropdown list. The Parent Scan Template dropdown appears.
9. Leave the **Parent Scan Template** set to **Host Range** in the dropdown list. A field is automatically generated for this value when selected.
10. In the **Fields** section, add a field for each of our script output object's properties by typing the following in the text box and clicking the **Add field** button:

Field Name	Parent Field
DistinguishedName	<None>
Name	HostRange
ObjectGUID	<None>

A field with the name value HostRange for the parent HostRange is automatically created and needs to be renamed to "Name":



11. When done, click the **Save** button.

Machines

Create the scan template to contain the output from the Machine Scanner script. This script takes the name of an OU retrieved from our previous step, scans that OU for computers, and returns a list of custom objects containing certain properties of each computer. In this tutorial we are capturing these properties:

- ADGUID
- ComputerName
- DistinguishedName
- DNSHostName
- OperatingSystem

If future scanners need more information about the computer, you can easily modify this script to return additional properties.

1. Click the **Create Scan Template** button. The New Scan Template page appears.
2. Type `PS Machine` in the **Name** text box.
3. Select the **State** checkbox to make the template **Enabled**.
4. Set the **Scan Type** dropdown list value to **Machine**.
5. The **Parent Scan Template** dropdown list automatically appears, set to **Computer**, leave it unchanged.
6. In the **Fields** section, two fields automatically appear, one for Machine which needs to be renamed to "ComputerName" and another for "OperatingSystem" which will be left as is.
7. Add a field for each of our remaining script output object's properties by typing the following in the text box and clicking the **Add field** button:

Field Name	Parent Field
ADGUID	<None>
ComputerName	Machine
DistinguishedName	<None>
DNSHostName	<None>
OperatingSystem	OperatingSystem

8. Click the **Save** button.

Local Accounts

Our Local Account Scanner script takes the computer name of a computer retrieved from the previous step, scans that computer for local accounts, and returns a list of custom objects containing the following properties from each account:

- Disabled
- Name
- Resource

The setup of these fields in the Local Account scan template is different than the other templates that we have created so far. The parent template for local accounts is "Account" and it has three fields: Username, Password, and Resource. Our script is not able to return the password on the account, so the objects returned do not have that as a property. We need to map this parent field to a field on our template, but it is only used internally.

1. Click the **Create Scan Template** button. The New Scan Template page appears.
2. Type `PS Account` in the **Name** text box.
3. Select the **State** checkbox to make the template **Enabled**.
4. Set the **Scan Type** to **Account** in the dropdown list.
5. The **Parent Scan Template** dropdown list automatically appears set to **Account (Basic)**. Leave it unchanged.
6. In the **Fields** section, three fields automatically appear. Leave Resource and Password as they are. Rename Username as shown below.

   Add a field for each of our remaining script output object's properties by typing the following in the text box and clicking the **Add field** button:

Field Name	Parent Field
Disabled	<None>
Name	Username
Password	Password
Resource	Resource

Dependencies Scan Template

The final scan template to set up is the one used to find Windows Service dependencies. The script will return a list of all Windows Services on a computer along with account information for that service. The properties returned by the script for each service are:

- AccountStatus
- DependencyType
- Domain
- Enabled
- Machine
- ServiceName
- Username

The complete the setup:

1. Click the **Create Scan Template** button. The New Scan Template page appears.
2. Type `PS Dependency` in the **Name** text box.
3. Select the **State** checkbox to make the template **Enabled**.
4. Set the **Scan Type** to **Dependency** in the dropdown list.

5. The **Parent Scan Template** dropdown list automatically appears set to **Computer Dependency (Basic)**. Leave it unchanged.
6. The **Account Scan Template** dropdown list automatically appears, set it to **PS Account**.
7. In the **Fields** section, four fields appear automatically, leave them all unchanged:
   - Machine
   - ServiceName
   - Username
   - Domain

   Add a field for each of our remaining script output object's properties by typing the following in the text box and clicking the **Add field** button:

Field Name	Parent Field
AccountStatus	None
DependencyType	None
Domain	Domain
Enabled	<None>
Machine	Machine
ServiceName	ServiceName
Username	Username

Task Four: Setting up Discovery Scanners and Sources

Discovery Scanners

Now that you have created the scan templates that our scripted discovery source will need, you can create the discovery scanners.

When creating a new scanner you must specify:

- Which step the scanner runs on.
- What type of base scanner to use (for example, Manual Input, Windows Discovery, or PowerShell Discovery).
- Which scan provides the input for the scan.
- Which scan template represents the output of the scan.
- When using a PowerShell base scanner, you also select what script to run and any arguments to pass to the script.

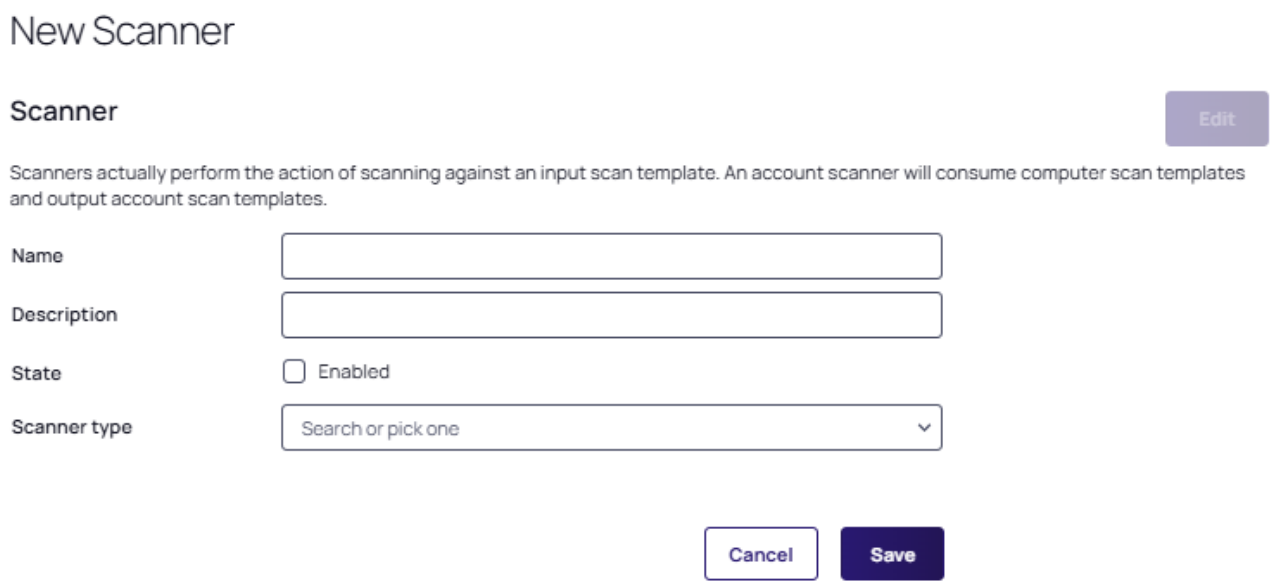
To get started:

1. Access **Discovery** and select the **Configuration** tab.
2. Click the **Discovery Configuration Options** dropdown and select **Extensible Discovery**.
3. Select the **Configure Discovery Scanners** button. The Scanners Definition page appears on the **Scanners** tab.

   The page is similar to the scan templates tab page, but with a list of configured scanners displayed. Secret Server comes with discovery scanners for each built-in scanner. Add a new PowerShell scanner for all four scanner types, using the scripts and scan templates we set up in the previous sections.

Host Ranges

1. Click the **Create Scanner** button. The New Scanner page appears:

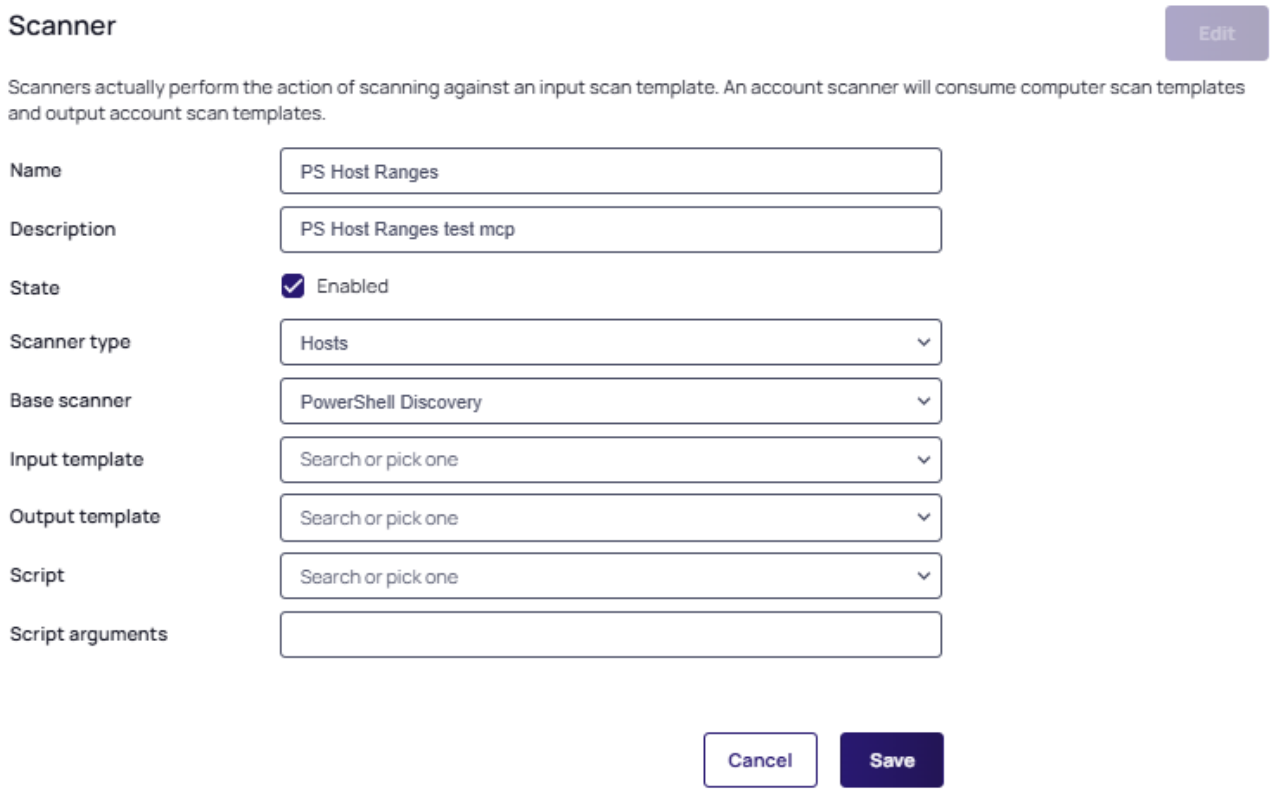


2. Type `PS Host Ranges` in the **Name** text box.
3. Type a description in the **Description** text box.
4. Select the **State** checkbox to make the scanner **Enabled**.
5. Set the **Scanner Type** dropdown list set to **Hosts**.

> Discovery scanning always proceeds in the following order: Host Ranges > Machines > Local Accounts > Dependencies. The scan templates, scanners, and source pages all organize their contents in the same order.

> Although the discovery scanning process proceeds in the order just mentioned, it is important to realize the output of each step *may* not be the input of the next step. Machines take host ranges as their input, and local accounts take machines as their input, but dependencies do not take local accounts as their input. Like local accounts, dependencies are on machines, so they also take machines as their input.

6. The **Base Scanner** dropdown list appears automatically, select **PowerShell Discovery**. A selection of template and other fields appears:



> For any scripted scanner, choose **PowerShell Discovery** as the **Base Scanner**. This tells the discovery process that this scanner is running a script. Other base scanner options are available based on the discovery type. If you do not need to run a script for a specific step of discovery but do need to use a custom scan template for the input, output, or both to create a specific workflow, you can choose an option other than "PowerShell Discovery".

7. In the **Input Template** dropdown list, select **Active Directory Domain**.

   The input and output templates are where you define the information flow for the discovery process. Each scanner uses the output of a previous step as its input. Each scanner returns a list of results as its output. The input template defines what to use as the input data for the scanner. The output template defines what is returned from the scan and used elsewhere. To see what scanner consumes the output of another given scanner, look for the one that has the same input template as the original scanner's output.

> You can have multiple scanners in each step with the same input template, but each scanner has to have a unique output template. When a scanner runs, it compares the results of the current scan with the results of the previous scan that was stored in the database. It updates any existing records, adds new records for new items, and removes any records that do not match items found during the current scan. If there were more than one scanner with the same output template, the second scanner would overwrite the results of the first scan, making it pointless. This is why each output template must be unique.

8. In the **Output Template** dropdown list, select **PS Organizational Unit**.

   This is the Host Range template we created in the previous section. Generally, each output template feeds a single scanner at the next level, but you can have multiple scanners using the same input template, with each using the results to find different things. For example, you could have two local account scanners defined that both use the input from the previous find machines step—one for finding Windows local accounts and the other for finding AD accounts that have rights on the computer. In turn, each scanner returns its results to its own output scan template—one creating Windows account secrets and the other creating Active Directory account secrets.

9. In the **Script** dropdown list, select **Host Range Scanner**, the first script you created in the previous section.

   The script runs for each object matching the input template, using the arguments in the next step, to return an object defined by the output template.

10. Type the following in the **Script Arguments** text box, separating each with a space: `$[1]$Domain $[1]$username $[1]$Password`.

    Script arguments can be a combination of literal values and tokens. When the script runs, these tokens are replaced with values from the input object and any privileged accounts associated with the scanner. Privileged accounts are assigned to scanners when the scanners are added to a discovery source. The table below lists the tokens that can be used as script arguments.

11. Click the **Save** button to save the scanner.

Table: Script Tokens

Token	Description
$target	A generic placeholder for the input object. This is not used when scanning for host ranges because there is no previous scanner input source. For machine scanners, $target refers to either the OU (for Active Directory discovery sources) or the IP address (for Unix and ESXi discovery sources) from the host range input. For local account and dependency scanners, $target is the name of the scanned computer.
$[x]$Username	The username of the xth privileged account associated with the scanner. Each scanner can have one or more privileged accounts associated with it. If you need to use the username of the first privileged account in your script, you would type in $[1]$Username. The second would be $[2]$Username and so forth. You can have as many privileged accounts as necessary.

Token	Description
$[x]$Password	Similar to $[x]$Username, this is the password of the xth privileged account associated with the scanner.
$[x]$Domain	Similar to $[x]$Username, this is the fully-qualified domain name of the xth privileged account associated with the scanner.

Machines

Once you set up one discovery scanner, the rest should be straight-forward:

1. Click the **Create Scanner** button. The New Scanner page appears.
2. Type `PS Machines Ranges` in the **Name** text box.
3. Type a description in the **Description** text box.
4. Select the **State** checkbox to make the scanner **Enabled**.
5. Set the **Scanner Type** dropdown list to **Machines**.
6. In the **Base Scanner** dropdown list that automatically appears, select **PowerShell Discovery**. A selection of template and other fields appears.
7. In the **Input Template** dropdown list, select **PS Organizational Unit**. This is the same as the output template from the last scanner.
8. In the **Output Template** dropdown list, select **PS Machine**. This is the Host Range template we created in the previous section.
9. In the **Script** dropdown list, select **Machine Scanner**. The script runs for each object matching the input template, using the arguments in the next step, returning an object defined by the output template.
10. Type the following in the **Script Arguments** text box, separating each with a space: `$target $[1]$domain`.
11. Click the **Save** button to save the scanner.

Local Accounts

Repeat the process for the local accounts scanner:

1. Click the **Create Scanner** button. The New Scanner page appears.
2. Type `PS Accounts` in the **Name** text box.
3. Type a description in the **Description** text box.
4. Select the **State** checkbox to make the scanner **Enabled**.
5. Set the **Scanner Type** dropdown list to **Accounts**.
6. In the **Base Scanner** dropdown list that automatically appears, select **PowerShell Discovery**. A selection of template and other fields appears.
7. In the **Input Template** dropdown list, select **PS Machine**. This is the same as the output template from the last scanner.

8. In the **Output Template** dropdown list, select **Account (Basic)**.
9. In the **Script** dropdown list, select **Local Account Scanner**. The script runs for each object matching the input template, using the arguments in the next step, returning an object defined by the output template.
10. Type the following in the **Script Arguments** text box, separating each with a space: `$target $[1]$username $[1]$Domain $[1]$Password`.
11. Click the **Save** button to save the scanner.

Dependencies

Repeat the process for the dependencies scanner:

1. Click the **Create Scanner** button. The New Scanner page appears.
2. Type `PS Windows Services` in the **Name** text box.
3. Type a description in the **Description** text box.
4. Select the **State** checkbox to make the scanner **Enabled**.
5. Set the **Scanner Type** dropdown list to **Dependency**.
6. In the **Base Scanner** dropdown list that automatically appears, select **PowerShell Discovery**. A selection of template and other fields appears.
7. In the **Input Template** dropdown list, select **PS Machine**. This is the same as the output template from the PS Machines scanner.
8. In the **Output Template** dropdown list, select **Computer Dependency (Basic)**.
9. In the **Script** dropdown list, select **Windows Service Dependency Scanner**.
10. Type the following in the **Script Arguments** text box: `$target`.
11. Click the **Save** button to save the scanner.

Discovery Sources

The final step is to create a discovery source and assign the discovery scanners we just made to it:

1. Select **Discovery > Sources**. The Discovery Sources tab loads.
2. Note the list of existing discovery sources.

> If you upgraded from an earlier version of Secret Server and have created an AD domain, a corresponding discovery source is displayed on this page. If discovery was not enabled on that domain it will appear in this list under the Disabled status.

3. Click the **Create** dropdown button and select the **Active Directory** type. The Create Active Directory discovery source page appears.
4. Type in values for the **Discovery source name**, **Fully Qualified Domain Name** (FQDN), and **Friendly Name** parameters.

> All parameters with asterisks are required.

5. Select the **State** checkbox to make the source **Enabled** for scanning.

> Active discovery sources are scanned at the defined discovery interval. If you have multiple discovery sources, the discovery source with the most un-scanned computers is scanned first.

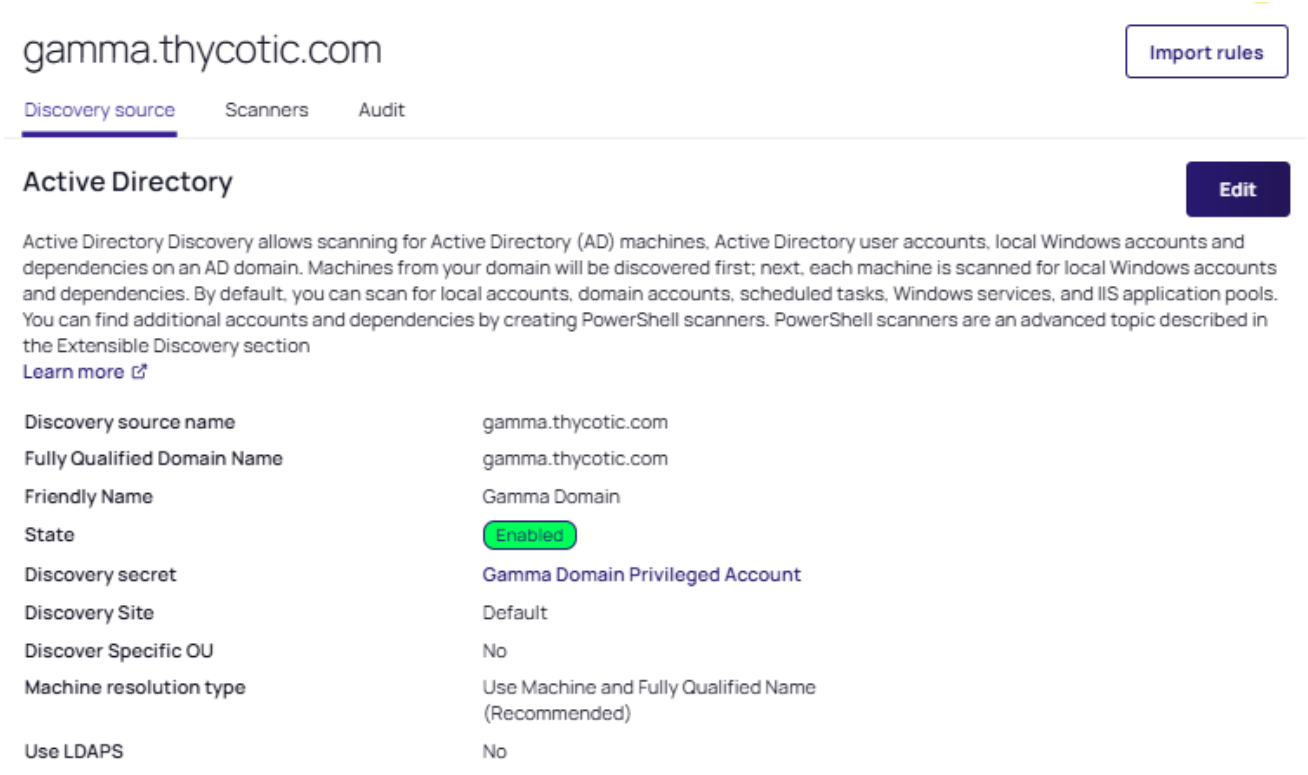
6. For the **Discovery secret** option click the **No Secret Selected** link. The Select Secret popup page appears.
7. Choose one of the following options:
   a. Option 1: Search for and select a secret that is used as the credentials for discovery scanning and AD synchronization. These credentials must have the proper rights to scan the remote machines. The popup page closes. The name of the secret you chose replaces the No Secret Selected link.
   b. Option 2: Create a new secret to be used as the credentials:
      i. Click the **Create New Secret** link. The Create New Secret page appears.
      ii. Search for and choose the **Generic Discovery Credentials** secret template. Another Create New Secret page appears.
      iii. Type or select the parameters needed for the discovery operation. Parameters with asterisks are required.
      iv. Click the **Create Secret** button.
8. In the **Discovery Site** dropdown list, select the desired site for the discovery source.

   If distributed engines are set up, the list shows all active sites. If no distributed engines are set up, the list defaults to local, and you cannot change it.
9. Select the **Discover Specific OU** checkbox to limit your discovery to an OU.

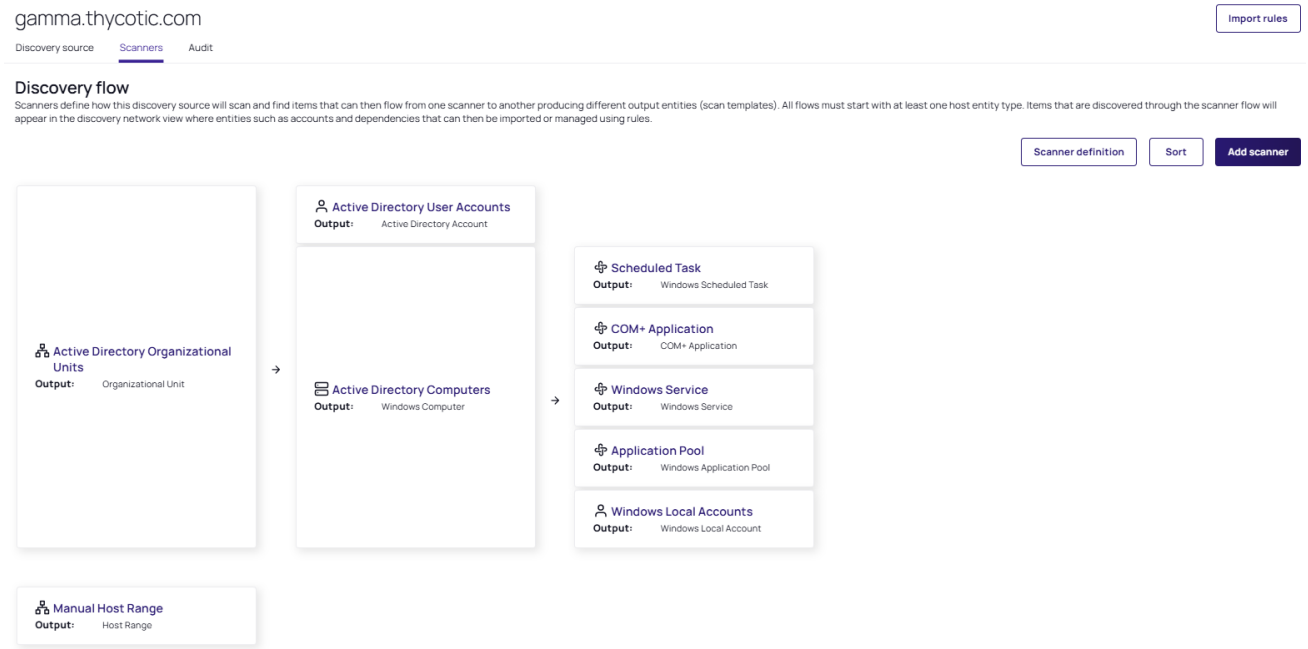
> See "Enabling Specific OU Domain Discovery" on page 568 to define the scanned OU. When you select this option, a Domain Scope tab appears on the Discovery Source page for the created AD discovery source.

10. Leave the **Machine Resolution Type** dropdown list set to **Use Machine and Fully Qualified Name** unless you have a specific reason to change it.
11. Select the **Use LDAPS** checkbox to use secure LDAP for the discovery.
12. Click the **Save** button.

    Secret Server attempts to access the domain with your specified credentials to ensure the configuration is correct. Secret Server must have access to the domain provided and the account credentials must work. Once the new discovery source is created it appears in the **Sources** tab of the Discovery page.
13. Click the newly created discovery source. The Discovery Source tab for the source appears:



14. Access the **Scanners** tab. The Discovery Flow page appears, for example:



> Note that a discovery source with default scanner options is already created. For this tutorial, we are not using any of those scanners.

15. Click on each scanner, and in their details panel, which opens on the right side of the screen, select **Remove Scanner**. In the verification popup, click the **Remove Scanner** button.
16. Click the **Add Scanner** button. The Add Scanner popup appears. It displays by default only scanners that can be added.
17. Set the **Scan Type** to **Hosts**.
18. Add the **PS Host Ranges** scanner.
19. Click on the **PS Host Ranges** scanner to open its details panel.
20. Click **Edit Scanner** to make it editable.
21. Click the **Add Secret** link. The Select Secret popup appears.
22. Select a secret that has permissions to scan the domain, such as the account you linked to the domain when adding the discovery source.
23. Click the **Save** button to save your settings, and exit out of the details panel.
24. Now that you have defined the host range scanner, repeat the process for the machine scanner, **PS Machines**, with the appropriate secret. Repeat any advanced settings from the last scanner.
25. Repeat the process for the local account scanner, **PS Accounts**, with the appropriate secret. Repeat any advanced settings from the last scanner.
26. Finally, repeat the process for the dependency scanner, **PS Window Services**, with the appropriate secret. Repeat any advanced settings from the last scanner.
27. Your scripted discovery source is now complete. You can go to the main discovery page to run discovery followed by a computer scan.
28. When both are done, you should see identical results in your discovery network view to what you would get if you ran discovery with our built-in scanners.

Manually Importing Local Accounts

Importing local accounts is the process of bringing in discovered local accounts for management by Secret Server:

1. Click **Administration** button on the side bar. The Secrets Administration page appears.
2. Click the **Discovery** link in the **Configuration > General** section. The Discovery Sources tab of the Discovery page appears.
3. Click the **Network View** button, which is not the same as the Network View tab. The Discovery Network View page appears. This page shows the computer accounts that have been found by Discovery. Clicking a domain name in the domain tree on the left displays the OUs available in that domain. Clicking on an OU displays the computers in that OU in the search grid.

> By default, this process uses the Import All discovery account rule.

4. Click a domain in the tree. The tree expands to show the domain's OUs.
5. Click an OU from that domain. The table on the right populates with the computers in that OU.
6. Click to select the computer accounts you want to import. Only accounts that have been assigned a secret can be selected. In fact, that is largely what import does—takes a discovered computer account and assigns it a secret so that Secret Server can manage it.
7. Select or fill in the scan template, secret type, folder for the new secrets, secret naming convention, and site (if this discovery uses distributed engines).
8. Click the **Next** button. The Password tab of the wizard appears.
9. Choose whether you have existing passwords for the accounts or wish to create new ones. If you choose the latter you can choose whether you want to manually create the new passwords or automatically create them based on the secret template's password settings.

   Remote password changing must be enabled to change the password. If that is the case, you will not see any selection buttons and a message appears instead.

10. Click the **Next** button. The Initial Takeover tab of the wizard appears.
11. If you chose to change the passwords, you need to select secrets to provide the initial password to do so—otherwise, discovery cannot access the account to change the password.
12. Click the **Next** button. The Password Changing tab of the wizard appears.
13. Select whether you want to use a secret credential or a privileged account to change the password in the future.
14. Click the **Finish** button. The final page of the wizard appears, and your choices are applied.
15. When the operation is complete, click the **Close** button. The Network View page reappears.
16. You can now see that the accounts you selected have secrets associated with them. You can click the secret name to go to that secret, which should indicate a successful heartbeat was conducted.

Platform-Specific Topics

Active Directory Discovery

Secret Server queries AD domains to obtain a list of Organizational Units (OUs) and Windows computers on the domain. These OUs and computers are recorded in the Secret Server database. Secret Server then attempts to connect to each computer and query for the following:

- **Domain Accounts:** AD user accounts
- **IIS Application Pools:** IIS application pools run by AD accounts
- **Local Accounts:** Local Windows accounts
- **Windows Services:** Windows services run by AD accounts
- **Scheduled Tasks:** Windows scheduled tasks run by AD accounts

Setting Permissions for Active Directory Scans

See "Account Permissions for Discovery" on page 528 for additional information about local account and dependency discovery.

The default configuration of Active Directory should allow you to scan computers and users. However, if your IT department has implemented security hardening measures, you may need to adjust your permissions accordingly.

Local Windows Accounts

The scanning account needs the "Access This Computer From the Network" permission (and possibly one more) on the endpoint:

1. Open the local group policy editor (gpedit.msc).
2. Go to **Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment**.
3. Double-click the **Access this computer from the network** policy. The properties for the policy appears.
4. Ensure the scanning account is one of the listed users. If not, click the **Add User or Group** button to add it.

   These settings apply only to the configurations listed below. Check the following list of the operating systems and updates to see if any match your system. If your system configuration is not in this list, then none of the settings above required to be set for you.

   - Windows 10, version 1607 and later
   - Windows 10, version 1511 with KB 4103198 installed
   - Windows 10, version 1507 with KB 4012606 installed
   - Windows 8.1 with KB 4102219 installed
   - Windows 7 with KB 4012218 installed
   - Windows Server 2022
   - Windows Server 2019
   - Windows Server 2016
   - Windows Server 2012 R2 with KB 4012219 installed

- Windows Server 2012 with KB 4012220 installed
- Windows Server 2008 R2 with KB 4012218 installed

> For more information on this security issue, see Network access: Restrict clients allowed to make remote calls to SAM.

Windows Services, Scheduled Tasks, App Pools, and COM+ Applications

> There are special considerations for discovering service accounts running COM+ Applications, please contact Delinea for more information.

> If you run discovery against Windows Server 2016 or 2019, scheduled tasks are not discovered unless your instance or engine are on the same domain as the target server. On Windows Server 2016 and up, scheduled task discovery only gets a security identifier (SID) for the user that runs the task. Secret Server has code to convert the SID to a username, but this only works if the code is being executed on the same domain as the scheduled task. If the SID cannot be translated, the scheduled task will not be saved with discovery.

To scan for service accounts, the account entered must be a domain account that is in the Administrators group on the target machines. Follow the instructions below in either case to ensure your account has the privileges to run a successful scan:

1. Open the group policy editor for your domain policy.
2. Go to **ComputerConfiguration** > **Preferences** > **Control Panel Settings**.
3. Right-click **Local Users and groups** and select **New** > **Local Group.**
4. Leave the **Action** dropdown list set to **Update**.
5. Click to select **Administrators (Built-in)** in the **Group Members** dropdown list.
6. Click the **Add…** button.
7. Search for the account you will use for discovery scanning.
8. Click the **OK** button to save your changes. The next time the group policy updates across your environment, the discovery account will be part of the local administrators group.
9. For strong security, configure the group policy to limit the logon privileges of that account:
    a. Open the group policy editor
    b. For your domain policy, go to **Computer Configuration** > **Policies** > **Windows Settings** > **Security Settings** > **Local Policies** > **User Rights Assignment**.
    c. Add your discovery account to the **Deny log on locally** policy.
    d. Add your discover account to the **Deny log on through Remote Desktop Services** policy.
    e. (Optional) Ensure the account is not part of the remote desktop users group.

Running and Interpreting Active Directory Discovery

This topic discusses how to configure, run, and interpret discovery scans on Active Directory systems. After the initial configuration, normally the discovery source is set to active, which runs the follow-on tasks automatically. You can also manually run either a discovery (locate computers on the domain) or computer (inspect the individual computers) scan.

Step One: Discovery Configuration

Running a discovery on an AD system is easy, assuming everything was configured correctly. To that end, follow these instructions first:

- "Setting Permissions for Active Directory Scans" on page 594
- "Creating a Discovery Source" on page 533
- "Enabling Specific OU Domain Discovery" on page 568 (optional)

Step Two: Discovery Scan

> When running discovery with a one way trust, make sure to use a user secret from the domain that has the universal trust.

When you complete the configuration and there is at least one active discovery source and discovery is enabled (the Active check box is selected), you can run a discovery scan manually or wait for an automatic one to start. A typical scan:

1. Runs discovery matching: The discovery matcher creates a link between existing active secrets and any existing secrets in Secret Server based on their machine names, accounts and dependencies. The matcher is automatic. When matches are found, the corresponding existing discovery results appear as "managed" in the discovery network view with a link to the existing secret or dependency.
2. Runs discovery rules: Secret Server attempts to match any unmanaged discovery results to the rule's parameters. If a rule matches the results, discovery automatically imports the results using the settings in the discovery rule. Once finished, discovery begins.
3. Runs the find host ranges scanner: The scanner (using the Windows discovery base scanner) runs with an Active Directory domain input template. The scanner determines which OUs are to be scanned and populates its organizational unit output template with a list of those OUs. The output template will be used by the following find machine scanner and also by the find local accounts scanner, which does not require machine information.
4. Runs the find machine scanner: The scanner (using the Windows Discovery base scanner) examines OUs from its organizational unit input template via LDAP and creates a list of machines with which it populates its Windows computer output template. This is the list of computers to run a dependency scan on. The find dependencies scanner uses this instance of the output template as its input template.

> You can see logs of this process by going to the Discovery Logs tab on the Discovery page.

To run a manual discovery scan, on the **Admin** menu, click the **Run Discovery Now** button and select **Run Discovery Scan**.

Step Three: Computer Scan

Once the computers in the desired AD domain or OU are discovered, a computer scan runs AD queries on each machine found during the discovery scan to attempts to the collect the information the discovery source was configure to collect, which can include local accounts, Windows services, scheduled tasks, and IIS application pools.

> If you run discovery against Windows Server 2016 or 2019, scheduled tasks are not discovered unless your instance or engine are on the same domain as the target server. On Windows Server 2016 and up, scheduled task discovery only gets a security identifier (SID) for the user that runs the task. Secret Server has code to convert the SID to a username, but this only works if the code is being executed on the same domain as the scheduled task. If the SID cannot be translated, the scheduled task will not be saved with discovery.

Specifically, the scan:

1. Runs the find local accounts scanner: Using the file load discovery base scanner, Secret Server examines OUs from its organizational unit input template via LDAP and creates a list of all AD admin accounts with which it populates its Active Directory account output template. This is the list of discovered admin accounts.
2. Runs the find dependencies scanner: Using the Windows discovery base scanner, Secret Server examines a list of machines from its Windows computer input template using various technologies. For example, application pools use Microsoft Web Administration (WMA) or, failing that, Windows Management Instrumentation (WMI). Services use WMI, and scheduled tasks use Windows' task scheduler interfaces. The find dependencies scanner can return any number of output templates as desired. These include: com+ application, computer dependency (basic), PS dependency, remote file, SQL dependency (basic), SSH dependency (basic), SSH key rotation dependency, Windows application pool, Windows scheduled task, and Windows service.

> You can see logs of this process by going to the Computer Scan Logs tab on the Discovery page.

To run a manual computer scan, on the **Admin** menu, click the **Run Discovery Now** button and select **Run Computer Scan**.

Step Four: Viewing Discovery Results

Browsing Discovery Results

1. Go to **Admin > Discovery**. The Discovery Sources tab of the Discovery page appears.
2. Click the **Network View** tab.

   The Discovery Network View page shows any discovered computer accounts. The domain tree on the left displays the domains as folders with OUs for that domain presented as folder contents. Clicking on a folder and then on an OU displays the computers in that OU in the table on the right.

   > For large numbers of domains you can type the domain name in the unlabeled search box over the domain folder tree and press <Enter> to narrow what domains are presented to you.

The discovery page has tabs for local account, service accounts, and domain or cloud accounts. All are very similar and draw from the same network tree on the left.

Searching Discovery Results

To search for a specific discovery source or OU, type the source or OU name in the search bar displayed at left. If results are found, click the result shown below the search field to highlight it. Now, only machines from that source or OU will be displayed at right.

To search for a specific computer name, account, or service name, type the search term in the search field on the right. Matching results are filtered below the search field.

To use advanced search settings, click the filter icon beside the search field. The filters panel appears.

Select an option in the **Filters** panel to match an account, computer, operating system, or rule.

> "Rule" only appears in the list box if discovery rules exist for local accounts. When you select it, another menu appears for selecting a rule. For more information about creating and searching with rules, see "Creating Discovery Rules" on page 540.

Click the **Managed** option buttons to select accounts managed or unmanaged by Secret Server.

Understanding Discovery Results

The table below describes the contents of each column:

Table: Discovery Results

Column	Description	Account Type (Local, Service)
Account	Username of discovered account.	Both
Computer	Computer name of the machine scanned. This is obtained from AD during the first part of the discovery process.	Both
Last Connected	Last date a user logged into the machine.	Local
Last Scanned	Last date that the machine was scanned by discovery.	Both
Org Unit	Organizational Unit the machine is joined to. This information is obtained from AD during the first part of the discovery process.	Local
Secret	If a secret name appears here, a credential secret already exists for the account listed in the account column. Otherwise, this column is blank.	Both

Column	Description	Account Type (Local, Service)
Service Name	Name of a discovered dependency.	Service
Status	Indicates that an account is managed by Secret Server, connectivity issues, or no accounts detected. For more information about error messages, see "Discovery Error Messages" on page 559.	Both
Type	Discovered dependency type icon. See the following table.	Service

Table: Service Account Dependency Types

Type	Icon	Service Name
Application Pool		IIS application pool name
Scheduled Task		Scheduled task name
Windows Service		Service name

To correctly identify and import IIS application pools for IIS 7 or higher, Secret Server requires a trust relationship between the scanned domain and domain that the Secret Server Web server is joined to.

AWS Account Discovery

Discovery must be enabled in Secret Server to discover AWS accounts.

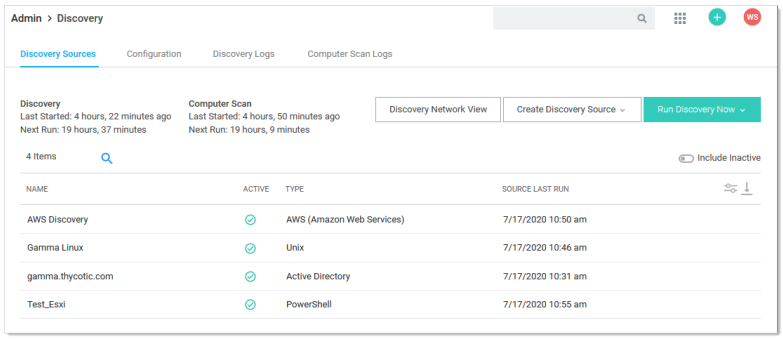
Secret Server can scan Amazon Web Services (AWS) for accounts that can access the cloud resource. Two types of secrets can be discovered and managed through SS:

- AWS Access Key: Keys used for programmatic integration with AWS.
- AWS Console Account: User login accounts for AWS.

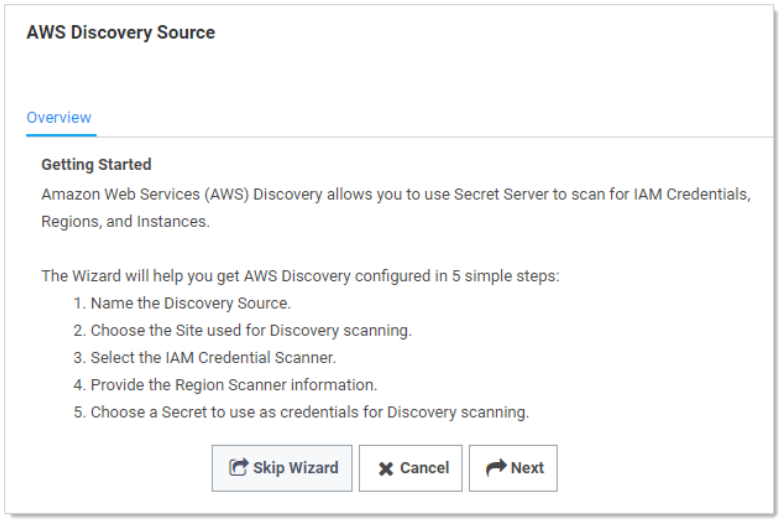
AWS Instance Discovery

Secret Server can now scan for instance resources in AWS. You can add this ability in the scanner settings section or through the wizard.

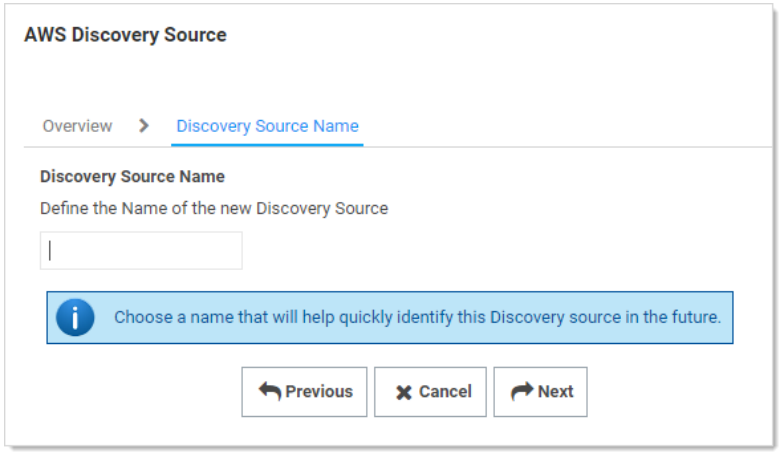
1. Create and AWS discovery source. See "Enabling AWS Discovery" on page 603.
2. Navigate to **Admin > Discovery**:



3. Click the **Create Discovery Source** dropdown list and select **AWS (Amazon Web Services)**. The AWS Discovery Source wizard Overview page appears:

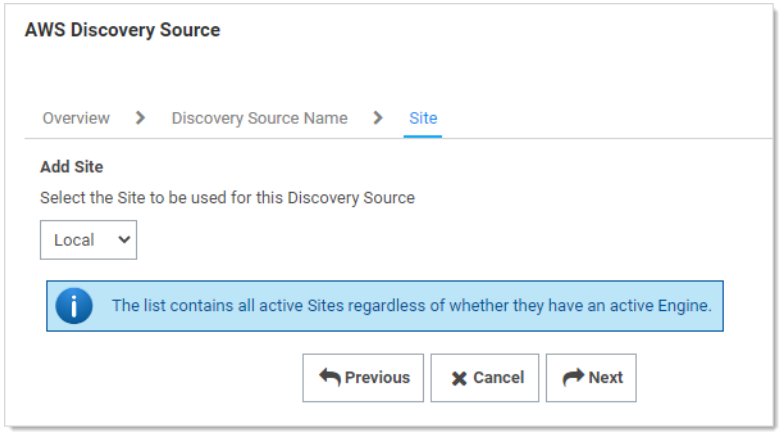


4. Click the **Next** button. The Discovery Source Name page appears:

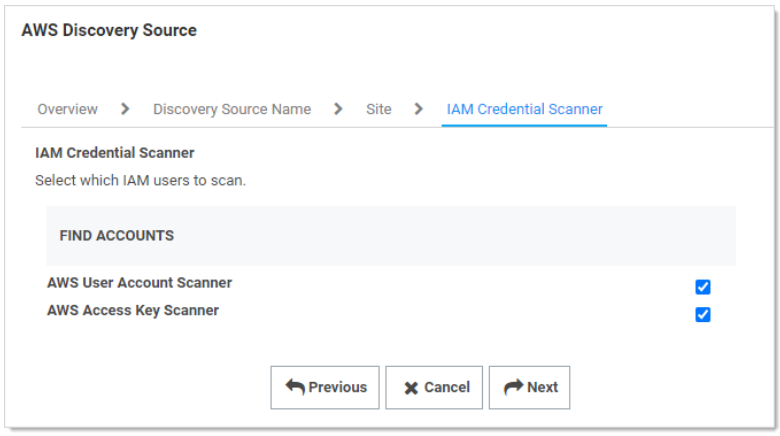


5. Type the name of the AWS discovery source in the **Discovery Source Name** text box.

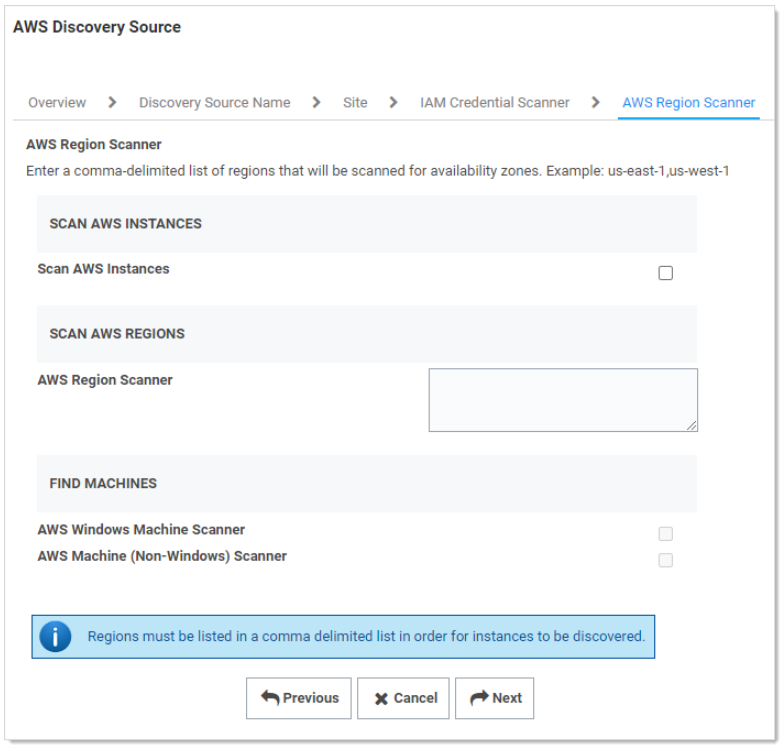
6. Click the **Next** button. The Site page appears:



7. Click the **Add Site** list box to select the site.
8. Click the **Next** button. AWS Service Account Scanner page appears:



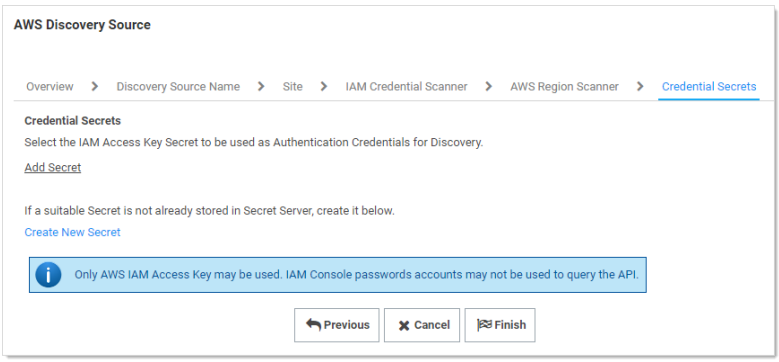
9. Click the check boxes for the scanners you desire.
10. Click the **Next** button.



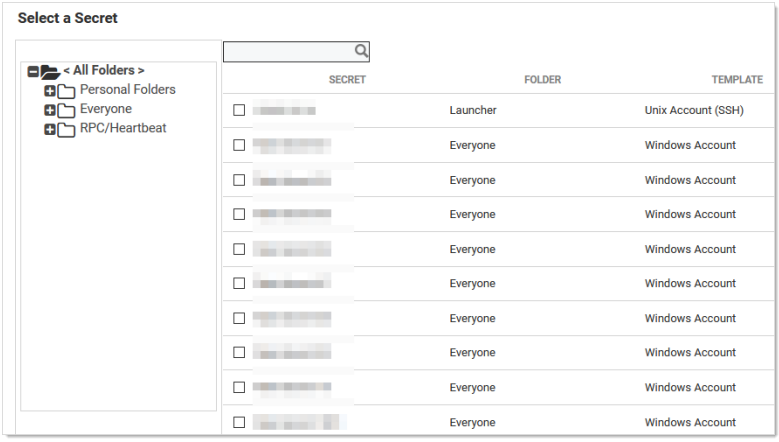
11. Click to select the **Scan AWS Instances** check box.
12. Type the regions you wish to scan for instances. The regions must be listed in a comma-delimited list for instances to be discovered.

    See Regions, Availability Zones, and Local Zones for more information on AWS regions.

13. Click to select the check boxes for the scanners you desire:
    - **AWS Windows Machine Scanner**: This is a machine scanner that scans each region and pulls all of the AWS Windows OS VM instances.
    - **AWS Machine (Non-Windows) Scanner**: This is a machine scanner that scans each region and pulls all of the AWS Non-Windows OS VM instances.
14. Click the **Next** button. The Credential Secrets page appears:



15. Click the **Add Secret** link. The Select a Secret popup appears:



16. Navigate the folder tree and select the secret you created earlier. As soon as you select the check box, the popup disappears and the secret appears under the Add Secret link.
17. Click the **Finish** button.

Enabling AWS Discovery

AWS Identity and Access Management (IAM)

1. For Secret Server to communicate with AWS, users with sufficient privileges need to create an access key for their account in AWS Identity and Access Management (IAM). The account used to do this requires the following permissions to discover users and access keys:
   - iam:ListUsers
   - iam:GetLoginProfile
   - iam:ListAccessKeys

   These permissions are limited to the resources the user is allowed to access.
2. Once this access key is created, use the access key and secret key to create a secret in Secret Server using the Amazon IAM key template.

3. Create a new AWS discovery source and use the Amazon IAM key as the credentials secret for that discovery source.

> AWS only allows programmatic integration through access keys. This type of secret is required for discovery to work. Discovery must be enabled in Secret Server for this feature to work.

Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (EC2) is a part of AWS that allows users to rent virtual computers on which to run their own computer applications. The EC2 account used to do this requires the following permissions to discover users and access keys:

- EC2:DescribeInstances
- EC2:DescribeAvailabilityZones

The configuration is very similar to IAM.

Password Management in AWS

Secret Server manages AWS key secrets through direct usage of the AWS IAM API. In order to do this, the following permissions are required.

Amazon IAM Keys

Password changing, privileged password changing, and running heartbeats are available for Amazon IAM key secrets. When an Amazon IAM key has its password changed through Secret Server, the new secret key is generated automatically and is not set by user input.

During password changing, you can disable or remove old keys through settings available in the advanced configuration:

- `<add key="ShouldDeletePreviousKey" value="true" />`
- `<add key="ShouldInactivatePreviousKey" value="true" />`

> Altering advanced settings can significantly impact the performance and behavior of Secret Server, so there is no direct link anywhere in Secret Server to the Advanced Settings page. If you need to change any advanced setting (as mentioned in this guide), please contact Delinea Technical Support.

Amazon IAM Console Password

Password changing, and privileged password changing are available for Amazon IAM console password secrets. Due to AWS IAM's restrictions on programmatic integration, this secret type cannot use Secret Server heartbeat.

In addition, an Amazon IAM key secret must be associated with an Amazon IAM console password secret for password changing to occur. To associate the two:

1. Create the Amazon IAM console password secret, and an Amazon IAM Key secret for an account that has the permissions to change the console user's password. This can be the console account's own access keys, if the

user has permission.

2. Navigate to the RPC tab of the Amazon IAM Console Password.
3. Under **Change Password Using Privileged Account** select **Edit** and choose the IAM key secret created in the previous step. RPC should now be possible on the console password secret.

Permissions Required for Secret Key Changes

These permissions are at the most granular level. You can implement broader methods through wildcard resource restrictions, permission policies, or groups.

Privileged Permissions: (those the AWS account needs to change another users' access keys):

- `iam:DeleteAccessKey` on resource `arn:aws:iam::<account>:user/<otherUserName>`
- `iam:UpdateAccessKey` on resource `arn:aws:iam::<account>:user/<otherUserName>`
- `iam:CreateAccessKey` on resource `arn:aws:iam::<account>:user/<otherUserName>`
- `iam:ListAccessKeys` on resource `arn:aws:iam::<account>:user/<otherUserName>`

Basic Permissions (those the AWS account needs to change its own access keys):

- `iam:DeleteAccessKey` on resource `arn:aws:iam::<account>:user/${aws:username}`
- `iam:UpdateAccessKey` on resource `arn:aws:iam::<account>:user/${aws:username}`
- `iam:CreateAccessKey` on resource `arn:aws:iam::<account>:user/${aws:username}`
- `iam:ListAccessKeys` on resource `arn:aws:iam::<account>:user/${aws:username}`

Permissions Required for Changing the Amazon IAM Console Password

These permissions are at the most granular level. You can implement broader methods through wildcard resource restrictions, permission policies, or groups.

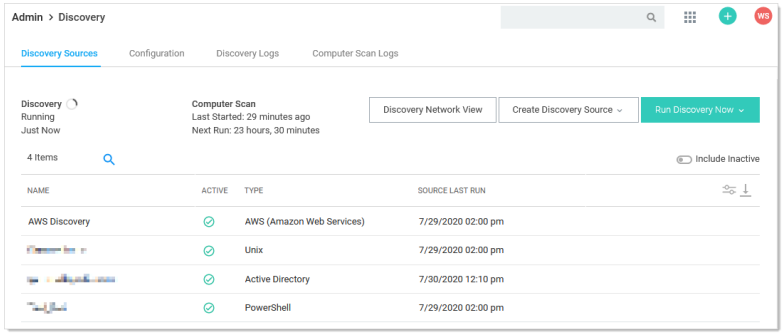
The permissions are:

- Privileged Permission: `iam:UpdateLoginProfile` on `resource arn:aws:iam::account>:user/<otherUserName>`
- Basic Permission: `iam:ChangePassword` on resource `arn:aws:iam::<account>:user/${aws:username}`

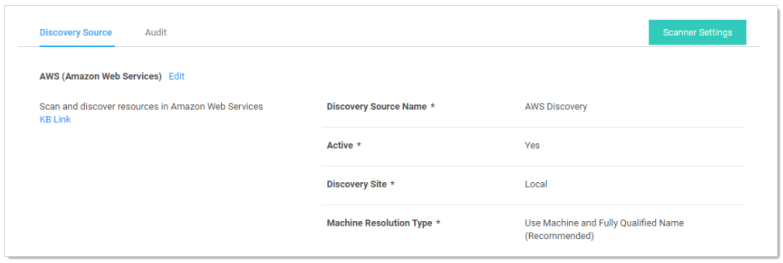
Viewing AWS Discovery Source Scanners

To view these scanners:

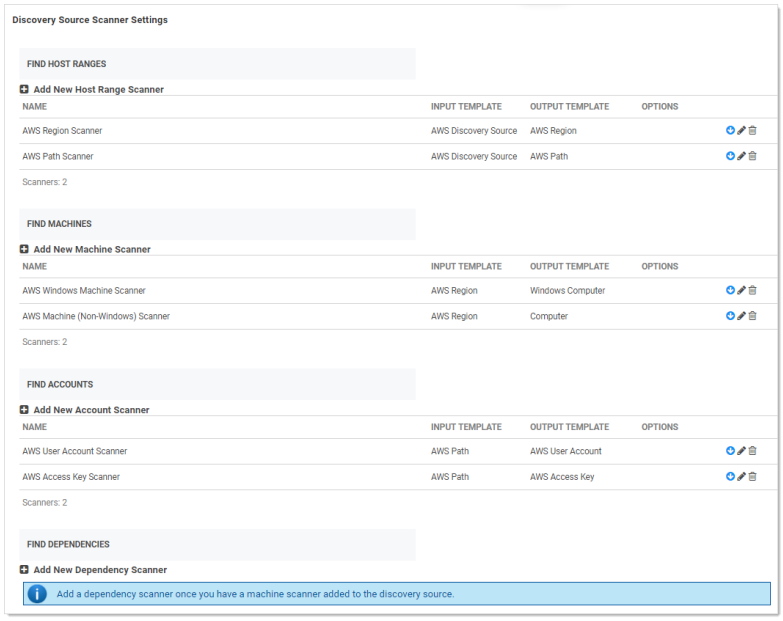
1. Go to **Admin > Discovery**.



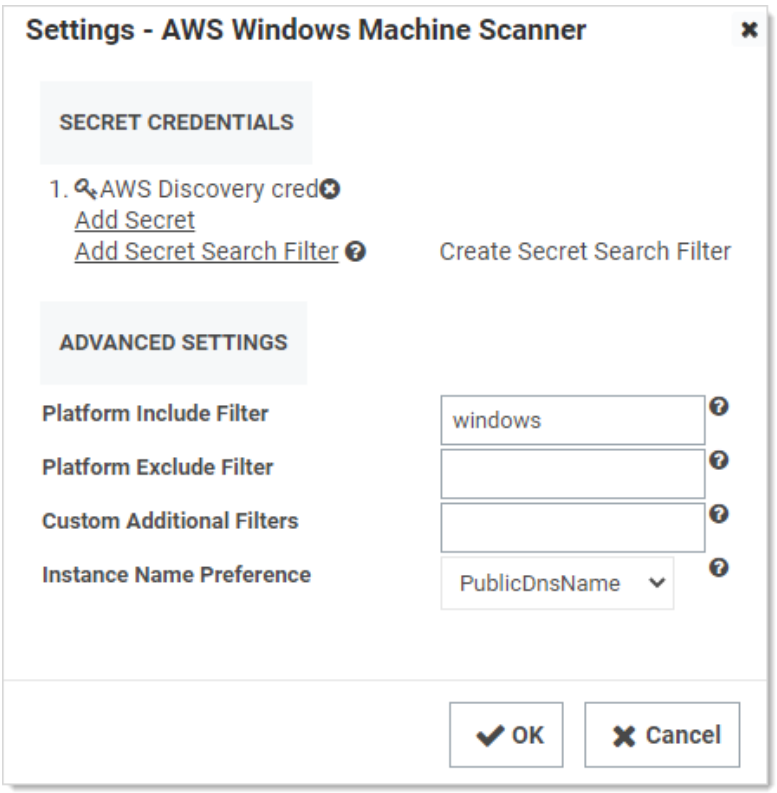
2. Click the discovery source name link in the table. The Discovery Source page for it appears:



3. Click the **Scanner Settings** button in the top right of the page. The Discovery Source Scanner Settings page appears, which lists the scanners.



4. Click pencil edit icon for the machine listing. The settings for that scanner appears:



5. Complete the following settings:
   - **Platform Include Filter**: Comma separated list for platforms to include in the scan. Example: `windows`.
   - **Platform Exclude Filter**: Comma separated list for platform to exclude from the scan. Example: `windows`,
   - **Custom Additional Filters**: Additional filters to scan. Example: `tag:Purpose=store,database;`
   - **Instance Name Preference**: If found on the instance, this is used for the Computer Name. Consider how the machine will be accessed with the selection. If selection is not found, it defaults to PrivateDnsName.
6. Click the **OK** button.

Entra ID Discovery

Discovery must be enabled in Secret Server to discover Entra ID accounts.

A distributed engine is required to use Entra ID Discovery with Secret Server Cloud. Secret Server On-Premises customers can leverage both distributed engines and webnodes.

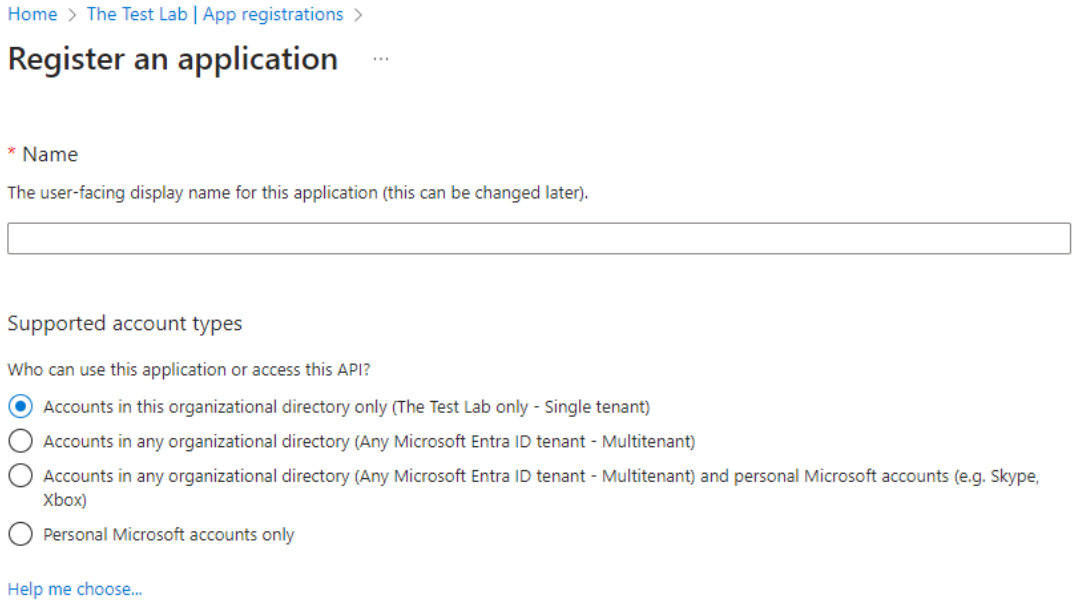
Overview

Secret Server can scan Microsoft Entra ID for Roles and Users. Users can be exported as secrets of the Entra ID User Account template.

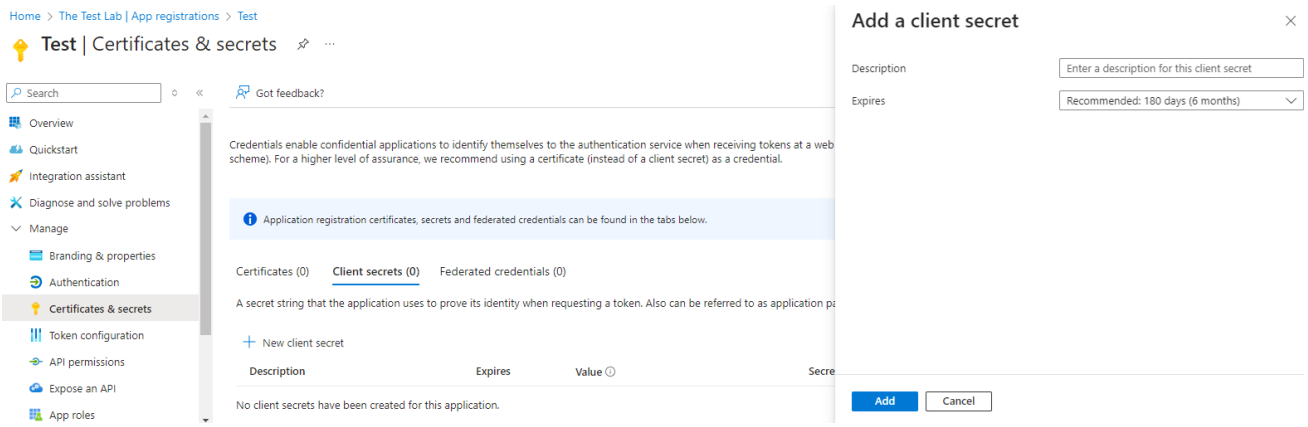
Configuration

Setting permissions:

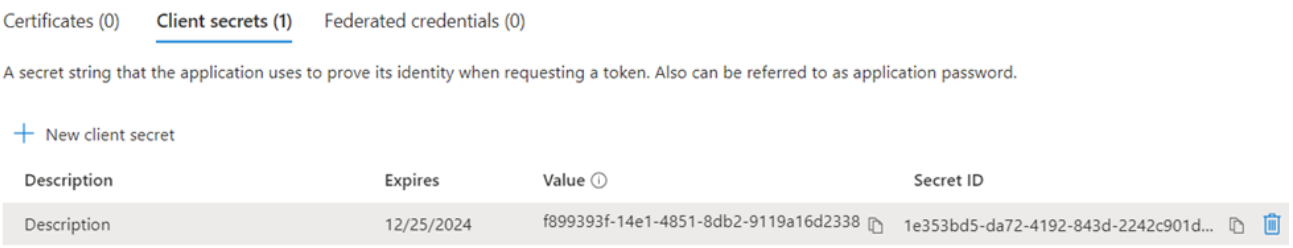
1. Create an App Registration in Entra ID:



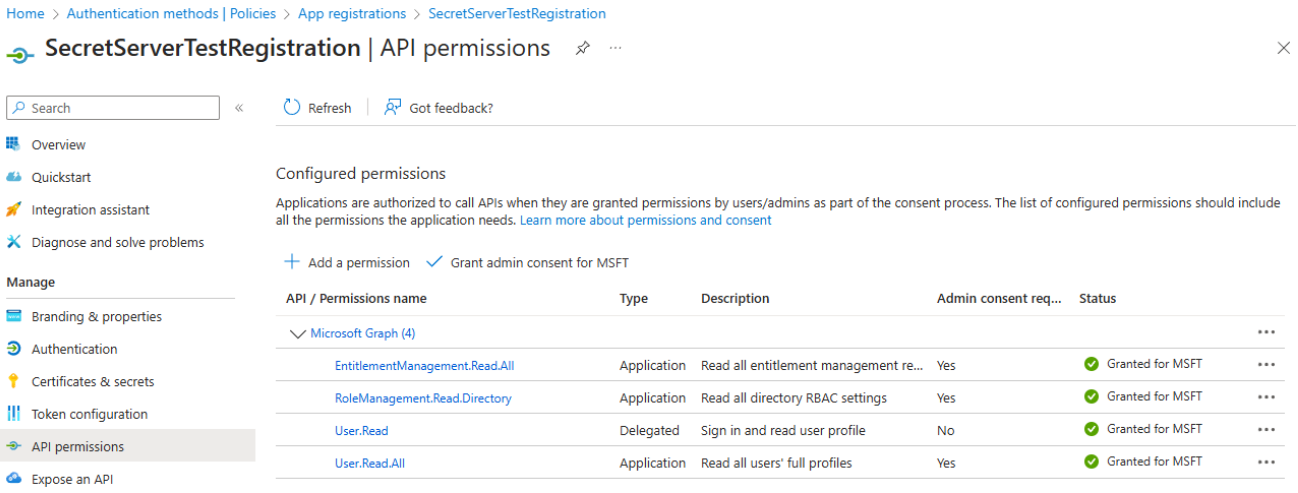
2. Add a client secret to the registration:



3. Once the client secret is created, the following data will appear. Copy the data in the **Value** column immediately–it will not be displayed again:



4. Grant API Permissions for the registration. Click **Add A Permission** to add each permission. The minimum required permissions for discovery are:
   - EntitlementManagement.Read.All
   - RoleManagement.Read.Directory
   - User.Read.All

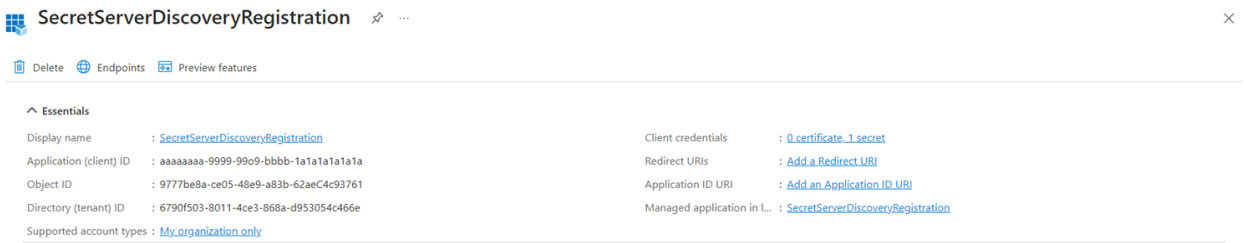


> Permissions must be added as Application Permissions not Delegated Permissions.

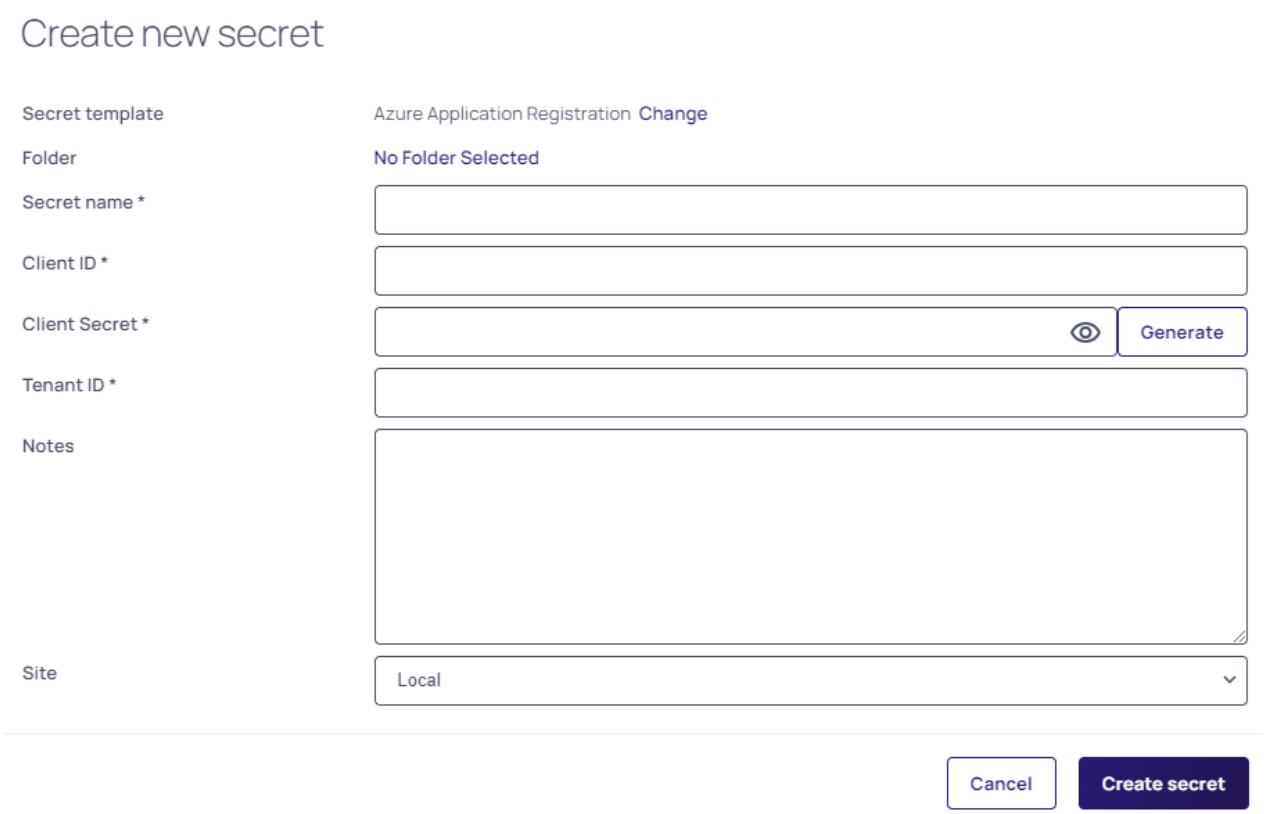
> The Application Registration may have permissions granted by default in addition to the ones listed above. These should have no impact on Discovery.

> If you want to use the same Application Registration for password changing you will also need the User.ReadWrite.All and UserAuthenticationMethod.ReadWrite.All permissions.

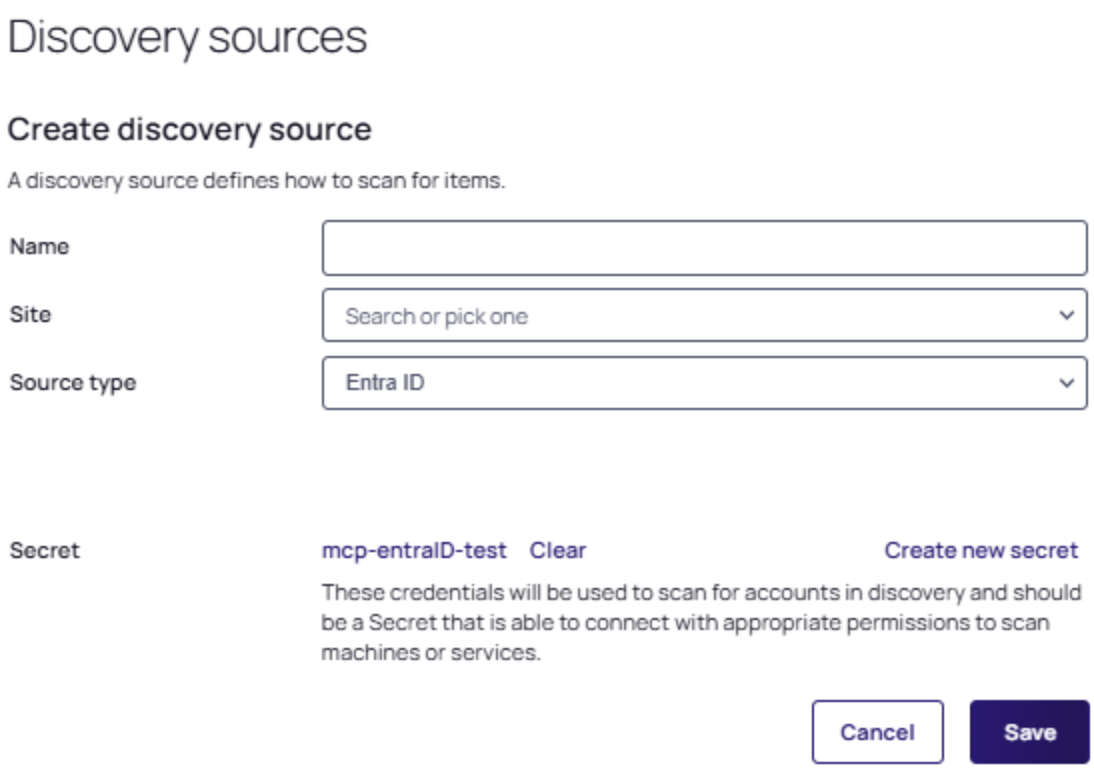
5. Map the appropriate fields to an Azure Application Registration secret in Secret Server:
   - Azure Portal:



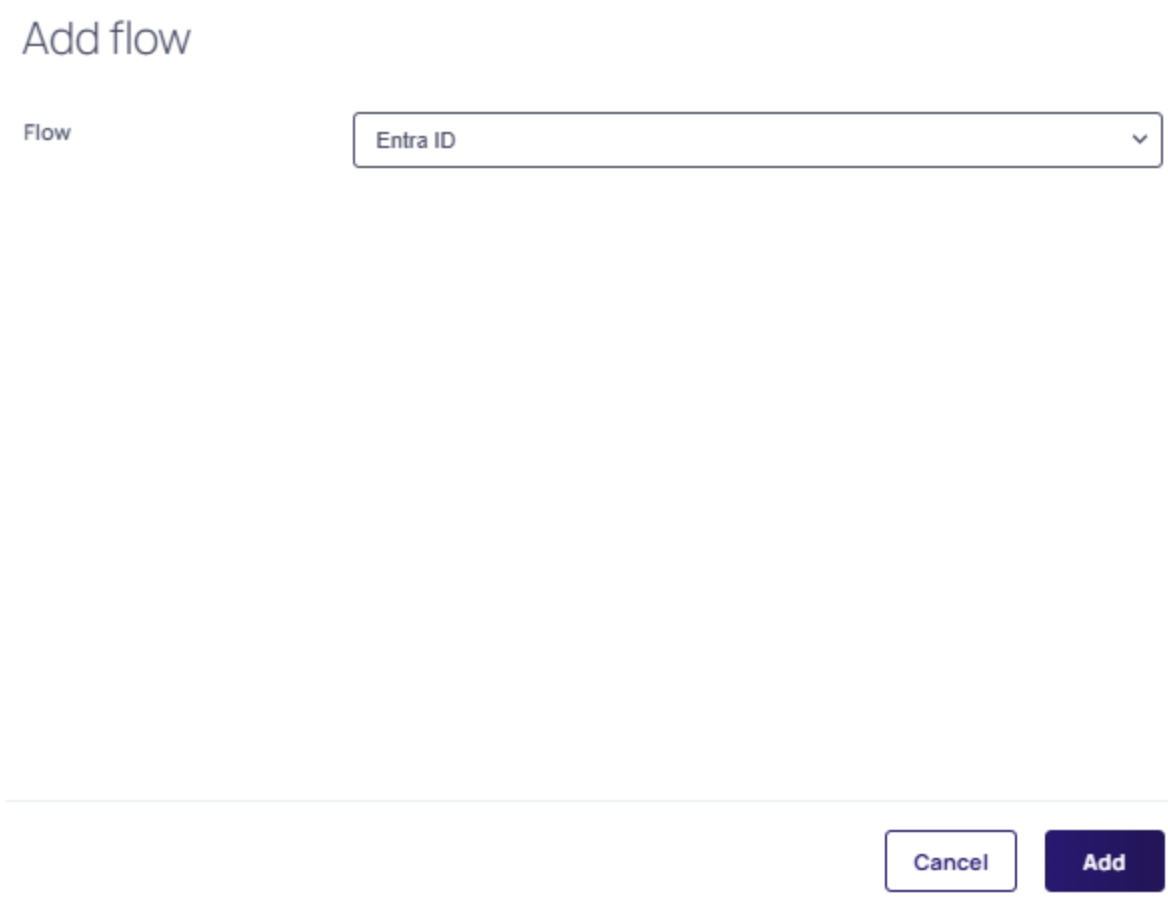
- Secret Server:



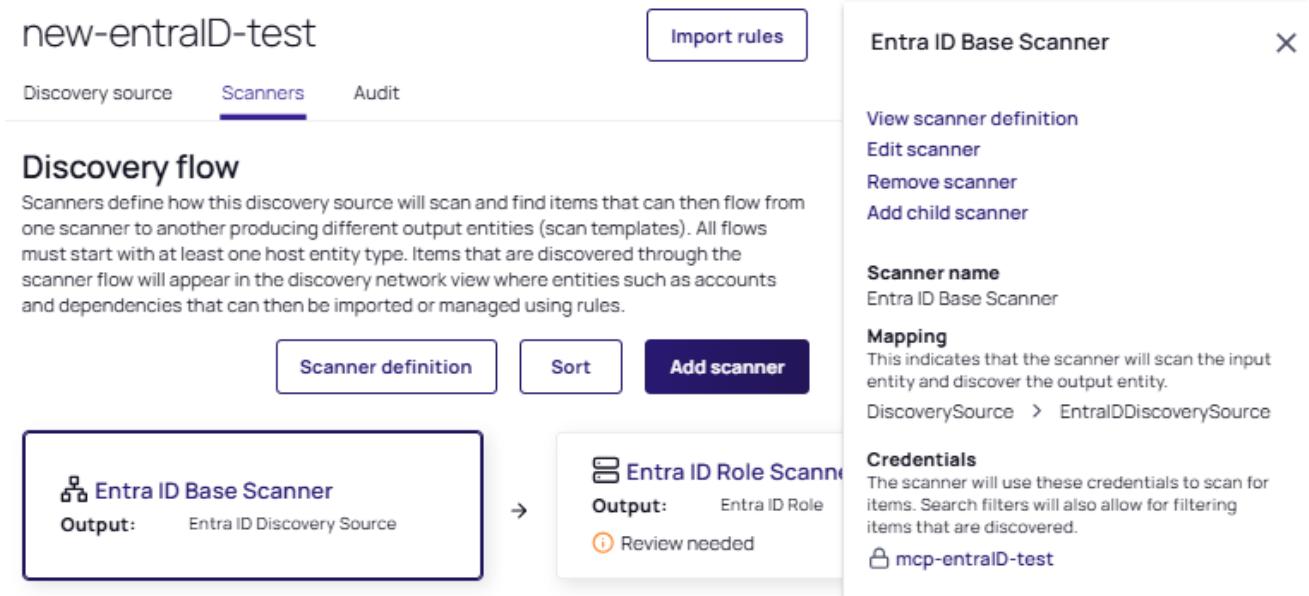
6. When mapping the fields from Entra ID to the new secret, note the following:
   - Application ID maps to Client ID.
   - Directory ID maps to Tenant ID.
   - The Client Secret is the value generated in Step 2.
7. In your Secret Server instance, create a new **Discovery Source** using the Azure Application Registration Secret:



8. By clicking **Save** you are automatically prompted to add the scanner flow for Entra ID to the Discovery Source:



9. For each scanner set the Azure Application Registration as the credential secret:



Scanning

Entra ID Roles can be obtained by running a Discovery Scan. To scan for Users, run a computer scan. Once scanning is complete you can view the role assignments for a user by clicking on their name in the **Discovery Network View** tab, or you can view user assignments for a role by clicking on the role name.

Local Account Discovery Methods

Remote Procedure Calls (RPC)

This is the method that is used for local account discovery for all versions of Secret Server prior to release 8.6.000000 and is the default for all upgrades and fresh installations. It uses the same technology as the Windows remote password changing in Secret Server and is the most dependable and proven of the options. It can, however, be slower in some environments when scanning computers over a WAN.

Windows Management Instrumentation (WMI)

This method uses the WMI technology to query the Windows computer. In some environments, this method can be faster than the Remote Procedure Call. It does, however, require having the proper permissions and network configuration setup correctly for WMI to run.

Attempt WMI First, and Failover to RPC if Needed

This option attempts to use the WMI method first, and if that fails to work correctly, it attempts the RPC method for local account discovery. This option is potentially slower because it has the possibility of performing two separate scans for each computer.

Google Cloud Platform Discovery

Overview

Secret Server can manage Google Cloud Platform (GCP) service accounts and VM instances. This feature allows users to run discovery to pull and manage VM Instances, as well as import and manage GCP service accounts.

> Accounts with Owner Permissions within Google's Admin panel cannot be granted API permissions by Google Design. Please create a new user account within Google for Secret Server's access into Google.

Configuration

Task 1: Creating GCP Service Accounts

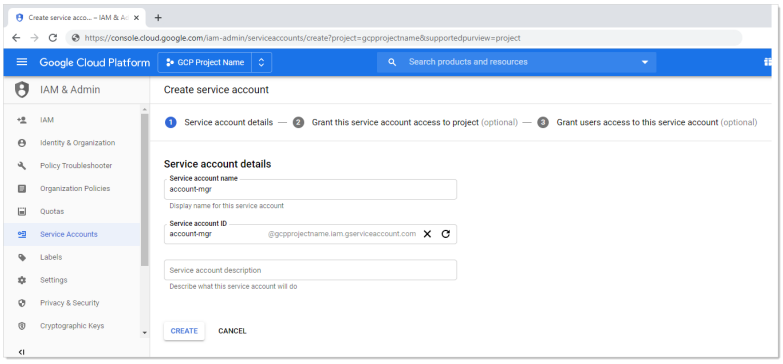
These are special accounts created in GCP to make authorized API calls for Compute Engine and other GCP applications.

> See GCP Service Accounts for more information.

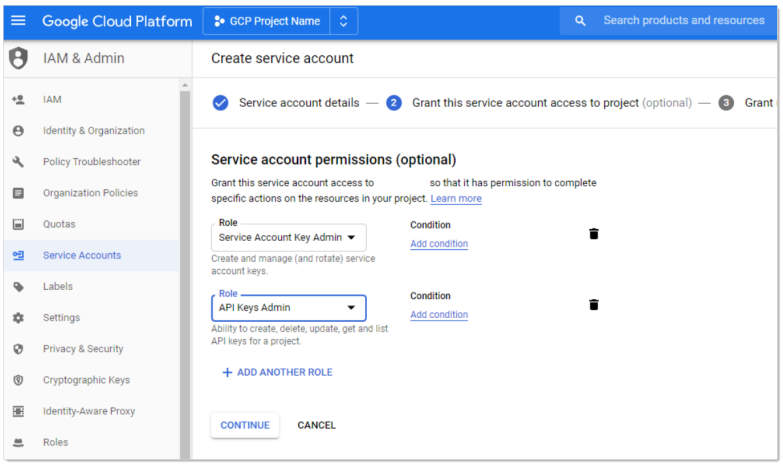
Secret Server uses the GCP service account to make authorized API calls to GCP to pull projects, zones, instances, service accounts and service account keys.

To create the service account:

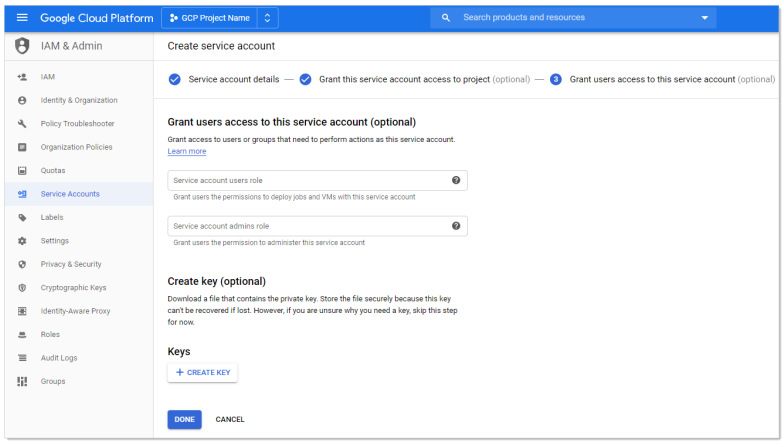
1. Click the **IAM & Admin** dropdown list in the left menu in GCP and select **Service Accounts**. A list of service accounts appears.
2. Click the **+ Create Service Account** button. The "Service account details" page of the Create Service Account wizard appears:



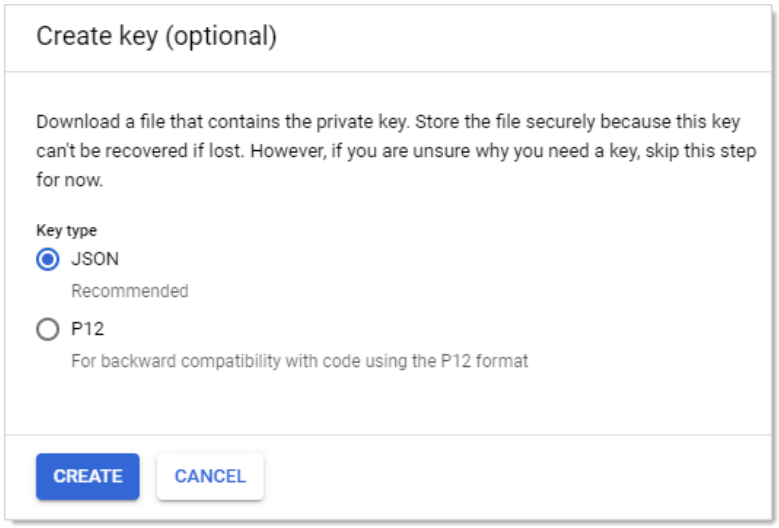
3. Type the service account name in the **Service Account Name** text box.
4. Start to type the service account ID name and select the service account in the **Service Account Name** text/list box.
5. Click the **Create** button. The "Grant this service account access to project (optional)" page appears:



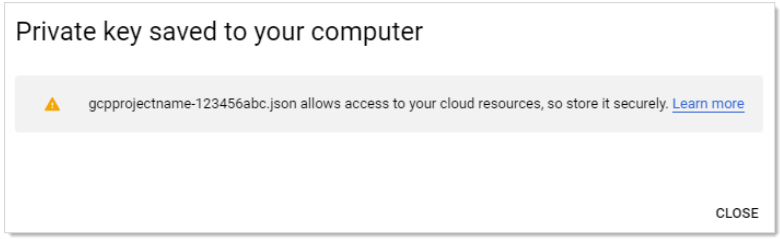
6. Click the **Role** list box and select **Service Account Key Admin**.
7. Click the **+ Add Role** button to add another role.
8. Click the new **Role** list box and select **API Keys Admin roles**.
9. Click the **Continue** button. The "Grant users access to this service account (optional)" page appears:



10. Click the **+ Create Key** button in the **Keys** section. The "Create key (optional)" popup appears:



11. Click to select the **JSON** selection button.
12. Click the **Create** button. This creates and downloads a JSON private key file. A confirmation popup appears:



13. Click the **Close** button in the bottom right. The service account is created, and its JSON private key is on your computer.

> Note where you downloaded the file. You will need it later in this instruction.

For more information on this process, see Creating and managing service accounts on the GCP website.

Task 2: Setting GCP Permissions

GCP permissions are IAM permissions from the IAM & Admin section of GCP. Without the proper permissions, GCP discovery, RPC, and heartbeat may not function properly.

For the service accounts to have access to a project, you must add the service account IAM permissions in each Project. If you did not add the permissions when you created the service account, you need to add the IAM permissions in the project they were created in as well.

Discovery

To run discovery in Secret Server, the GCP service account needs the "project viewer" read only permission, which can list projects, zones, service accounts, and instances.

To add the permission In GCP:

1. Click the **IAM & Admin** dropdown list in the left menu in GCP and select **IAM**. The "Permissions for project…" page appears.
2. Click the **Add** button. The "Add member to…" page appears.
3. Type the service account email address in the **Members** text box.
4. Click the **Roles** dropdown list to select **Project > Viewer** (you can also type it).
5. Click the **Add** button. The new member appears in the table on the "Permissions for project…" page.

RPC/Heartbeat

To run RPC/Heartbeat in Secret Server, the service account needs the "service account key admin" permission, which can create, delete, and rotate service account keys.

To add the permission In GCP:

1. Click the **IAM & Admin** dropdown list in the left menu in GCP and select **IAM**. The "Permissions for project…" page appears.
2. Click the **Add** button. The "Add member to…" page appears.
3. Type the service account email address in the **Members** text box.
4. Click the **Roles** dropdown list to select **Service Account Key Admin** (you can also type it).
5. Click the **Add** button. The new member appears in the table on the "Permissions for project…" page.

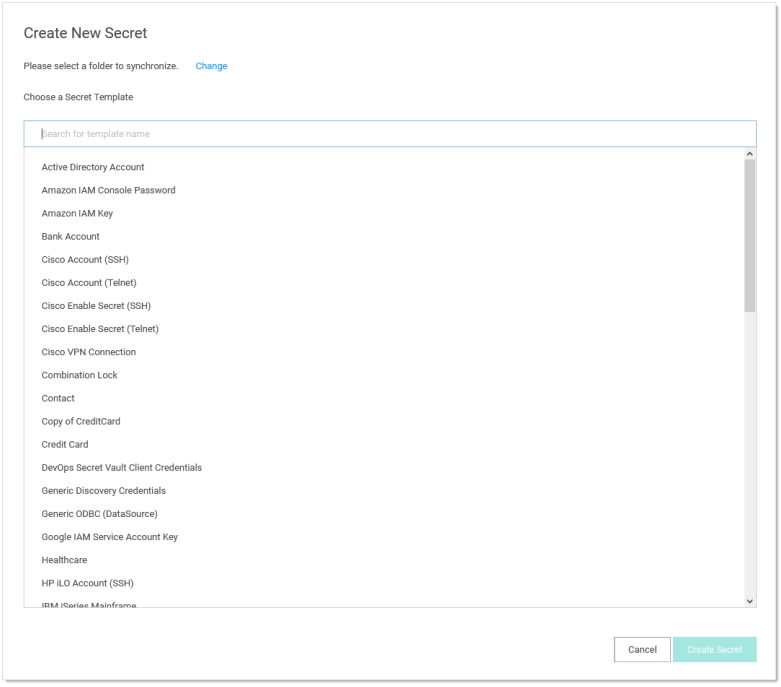
Task 3: Creating a GCP IAM Service-Account Secret

Secret Server now has a build in GCP IAM Service Account Key template.

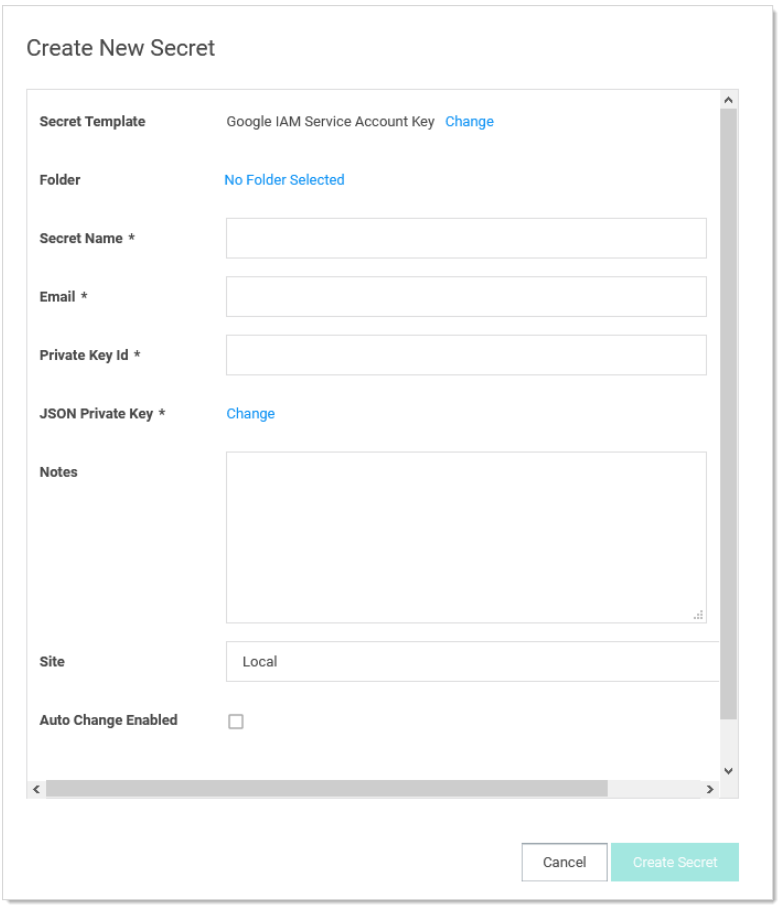
To create a Secret using GCP IAM service account key template, you must have the service account's JSON private key file from GCP (created earlier).

Create a new secret (see "Creating Secrets" on page 1127 for details):

1. Click the **+** on the **Secrets** item on the main menu. The "Create New Secret" page appears:



2. Select **Google IAM Service Account Key** as the template. Another "Create New Secret" page, tailored to GCP, appears:



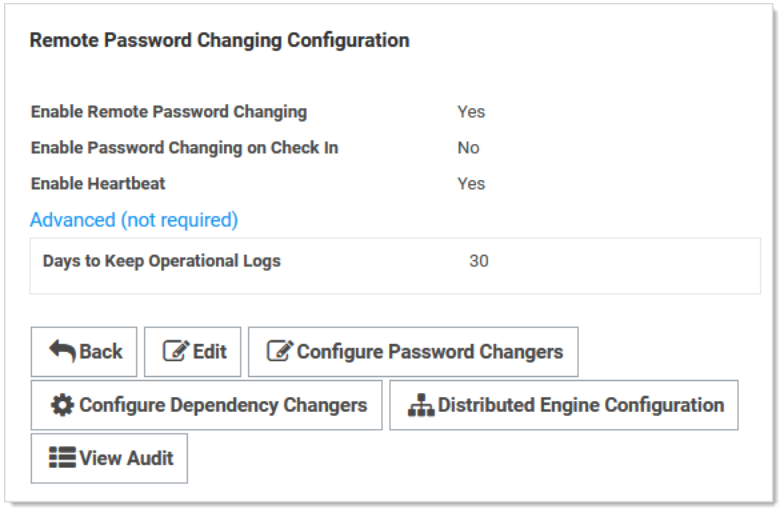
3. Click to select a folder for the new secret.
4. Type the secret's name in the **Secret Name** text box.
5. Type the service account email address (use client_email from the JSON private key file) in the **Email** text box.
6. Type the private key ID (use private_key_id from the JSON private key file) in the **Private Key ID** text box.
7. Click the **Change** button to upload the JSON private key file you created earlier.
8. Click the **Create Secret** button.

Task 4: Creating an RPC/Heartbeat Password Changer

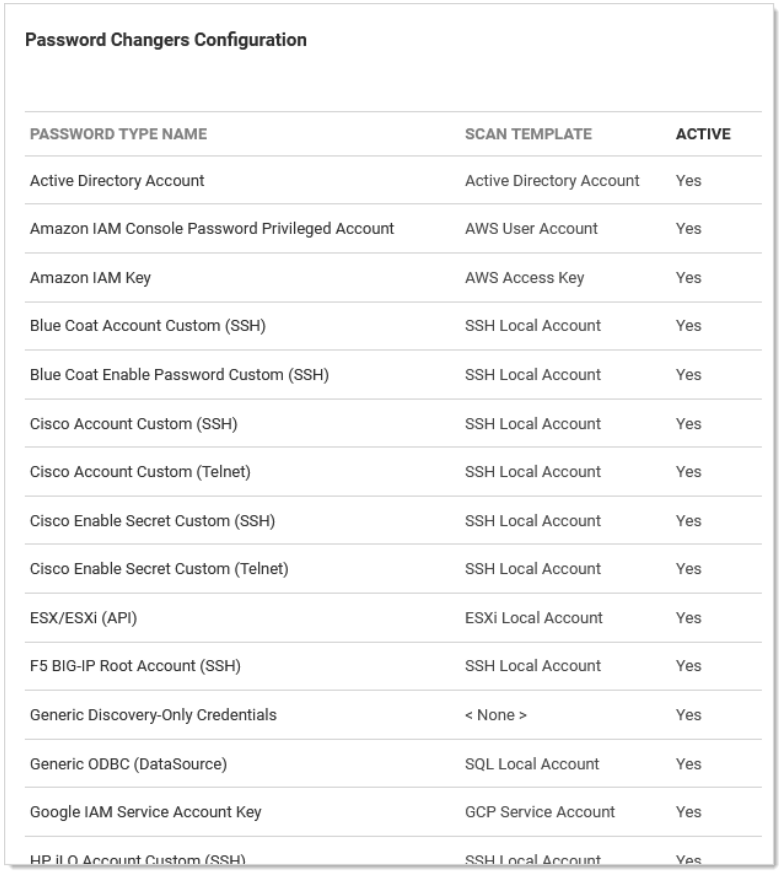
Secret Server can check if a service Account key is valid and can rotate the Service Account key. This should work the same as any other RPC or Heartbeat. RPC and Heartbeat must be enabled

RPC/Heartbeat can be tested from the Password Changers page

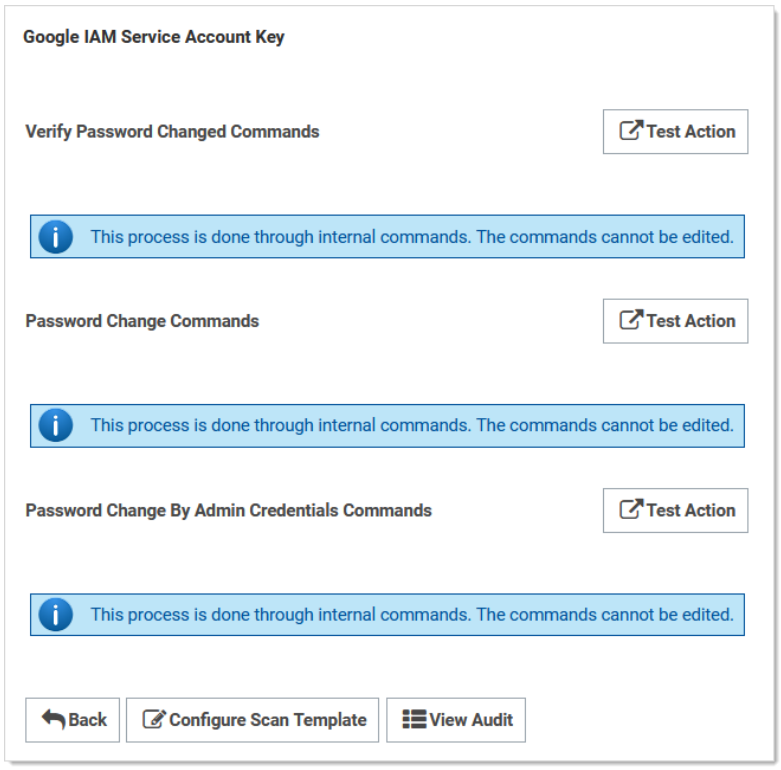
1. In Secret Server, go to **Admin > Remote Password Changing**:



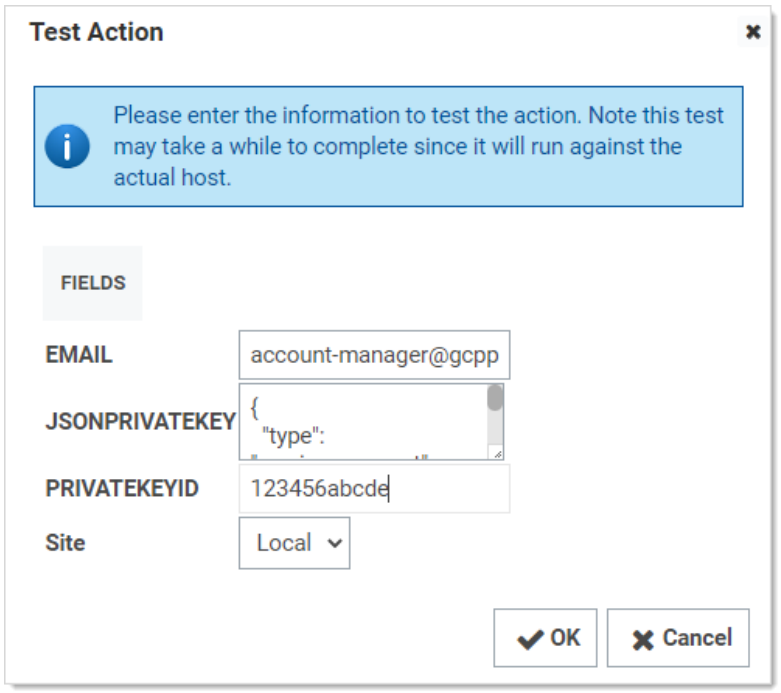
2. Click the Configure Password Changers button. The Password Changers Configuration page appears:



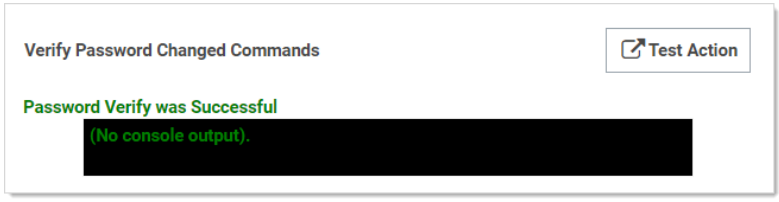
3. Click the **Google IAM Service Account Key** link. The "Google IAM Service Account Key" page appears:



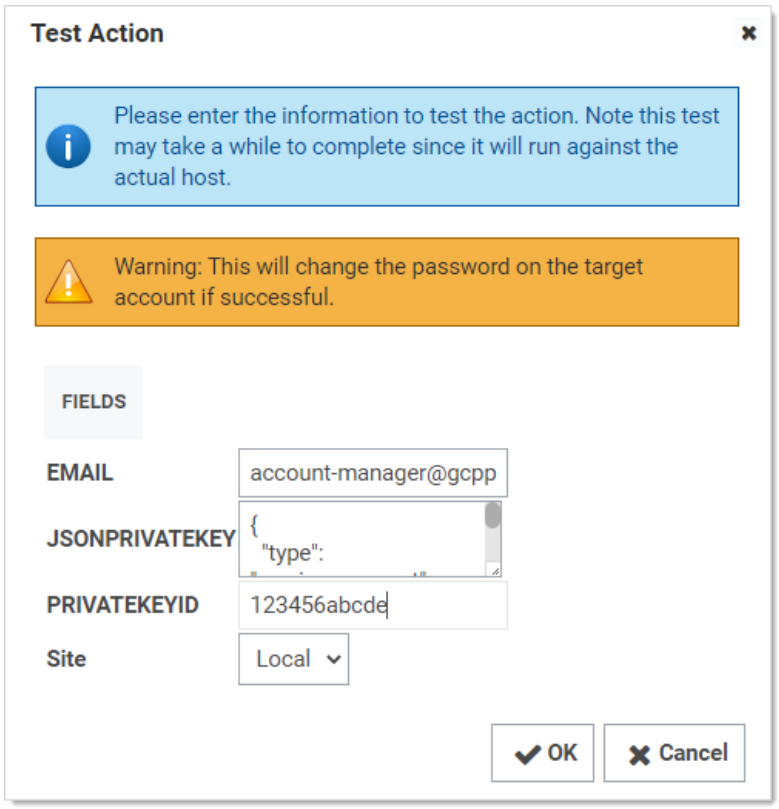
4. Test the heartbeat: Click the **Test Action** button in the **Verify Password Changed Commands** section. The Test Action popup appears:



5. Ensure that the **JSONPRIVATEKEY** text box is populated. The others are optional.
6. Click the **OK** button. The popup goes away. If successful, this appears on the previous page:

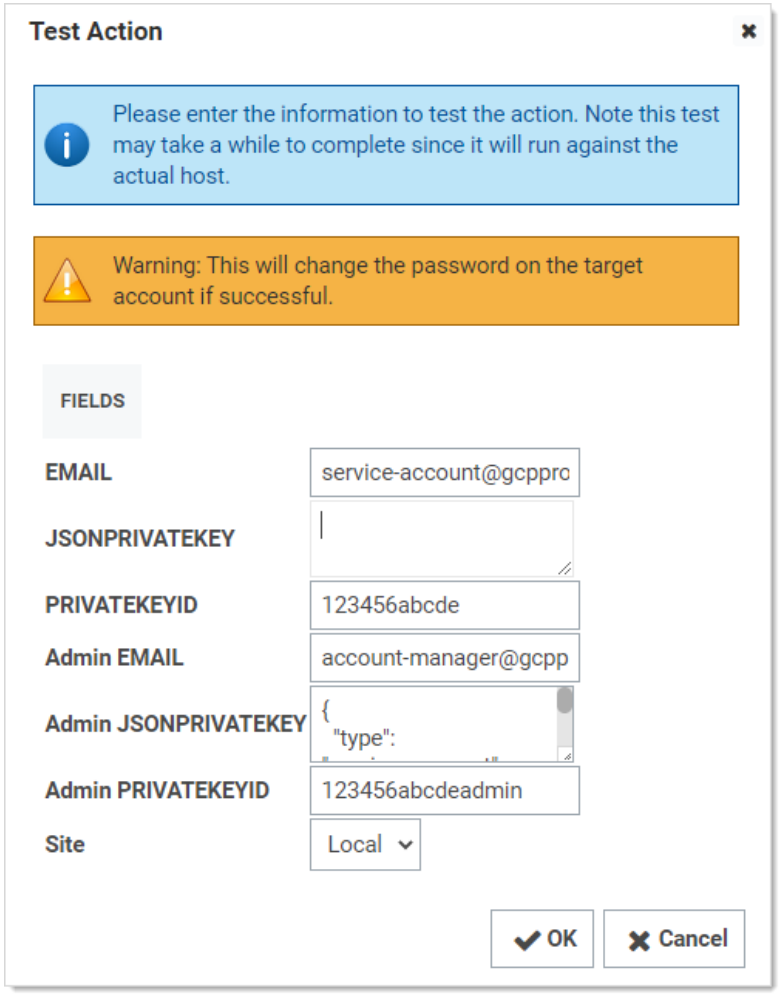


7. Test RPC: Click the **Test Action** button in the **Password Change Commands** section. The Test Action popup appears:

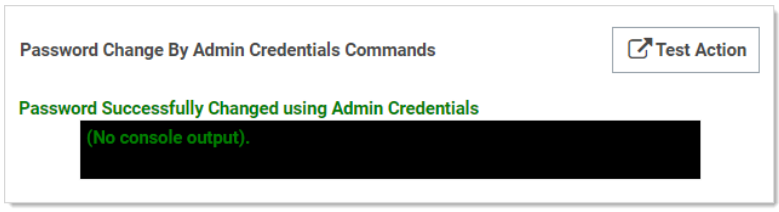


8. Ensure that the **JSONPRIVATEKEY** and **Email** text boxes are populated. The others are optional.
9. Click the **OK** button. The popup goes away. If successful, this appears on the previous page:

10. Test RPC with admin credentials: Click the **Test Action** button in the **Password Change By Admin Credentials Commands** section. The Test Action popup appears:



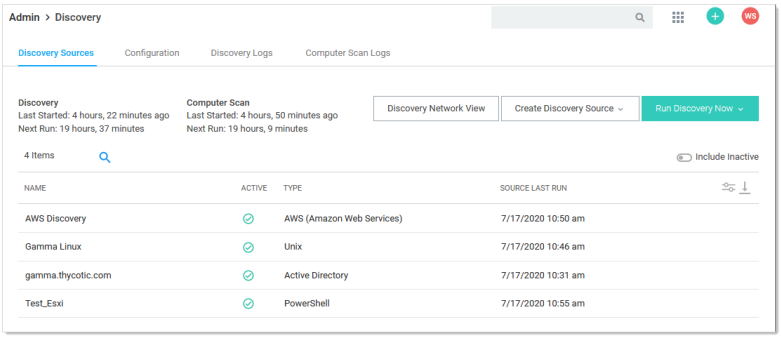
11. Ensure that all text boxes are populated except **JSONPRIVATEKEY**, **Admin Email**, and **Admin PRIVATEKEYID**, which are optional.
12. Click the **OK** button. The popup goes away. If successful, this appears on the previous page:



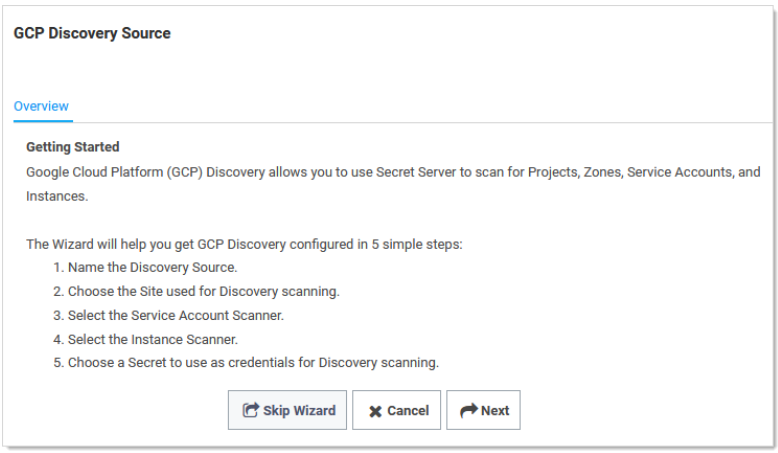
Task 5: Creating a GCP Discovery Source

Secret Server now has a built-in GCP discovery source wizard that creates the scanners to pull the projects, zones, service accounts. To create a GCP discovery source:

1. In Secret Server, go to **Admin > Discovery**:



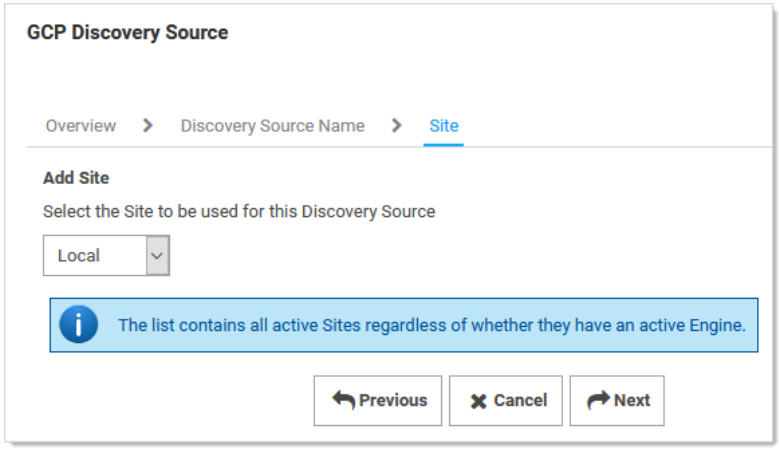
2. Click the **Create Discovery Source** dropdown list and select **GCP (Google Platform)**. The GCP Discovery Source wizard Overview page appears:



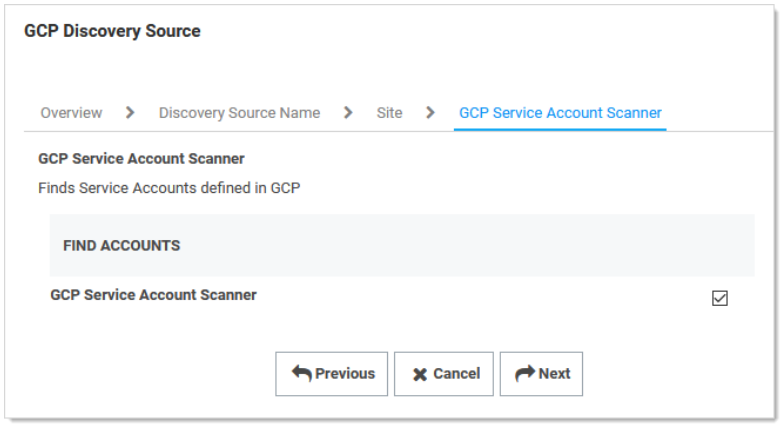
3. Click the **Next** button. The Discovery Source Name page appears:

4. Type the name of the GCP discovery source in the **Discovery Source Name** text box.

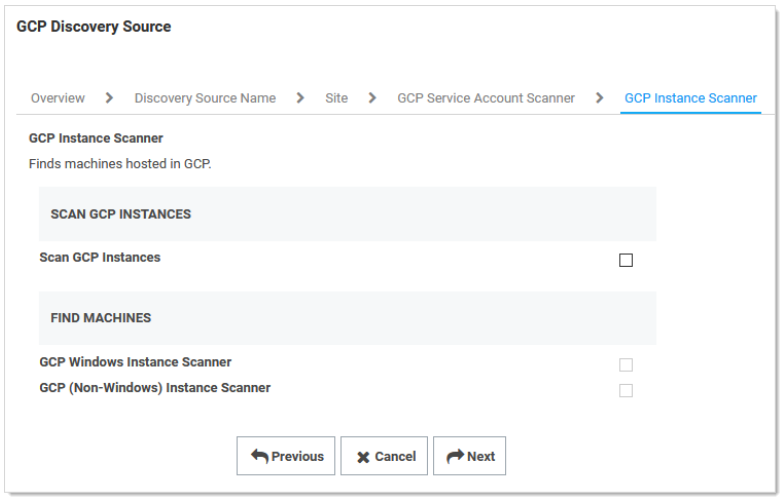
5. Click the **Next** button. The Site page appears:



6. Click the **Add Site** list box to select the site.
7. Click the **Next** button. GCP Service Account Scanner page appears:



8. Click the **Next** button.

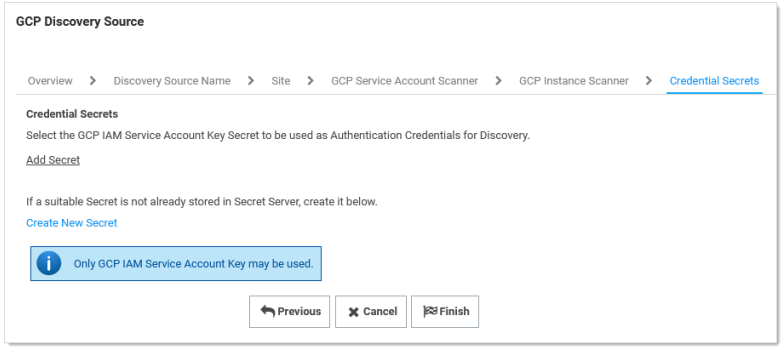


9. Click to select the **Scan GCP Instances** check box.
10. Click the check boxes for the scanners you desire. Currently, there are four discovery scanners for the GCP discovery source.

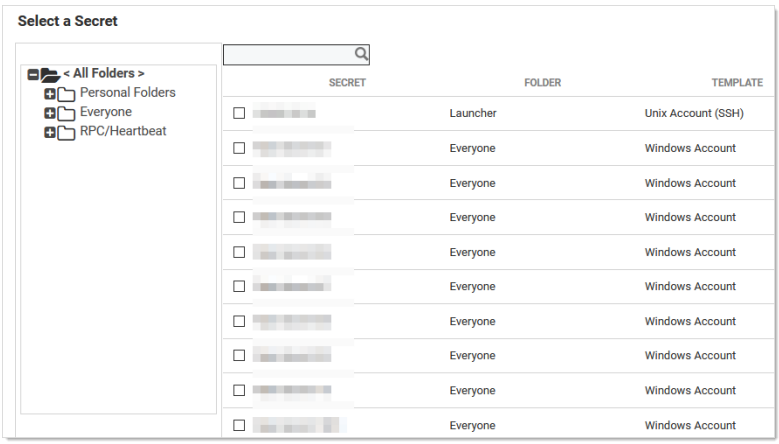
> In the future, we may add an Instance Local Account and a Service Account Dependency scanner.

- **GCP Project Scanner**: This is a host range scanner that scans the GCP and pulls all of the projects that the provided GCP service account secret has access to.
- **GCP Windows Instance Scanner**: This is a machine scanner that scans each project and pulls all of the GCP Windows OS VM instances.
- **GCP (Non-Windows) Instance Scanner**: This is a machine scanner that scans each project and pulls all of the GCP Non-Windows OS VM instances.
- **GCP Service Account Scanner**: This is an account scanner that scans each project and pull all of the GCP Service accounts.

11. Click the **Next** button. The Credential Secrets page appears:



12. Click the **Add Secret** link. The Select a Secret popup appears:



13. Navigate the folder tree and select the secret you created earlier. As soon as you select the check box, the popup disappears and the secret appears under the Add Secret link.
14. Click the **Finish** button.

Viewing Discovery Scanners for the GCP Discovery Source

To view these scanners:

1. In Secret Server, go to **Admin > Discovery**:
2. Go to **Admin > Discovery**.
3. Click the discovery source name link in the table. The Discovery Source page for it appears.
4. Click the **Scanner Settings** button in the top right of the page. The Discovery Source Scanner Settings page appears, which lists the scanners.

Instance Custom Filter

This option is only available for the instance scanners. The Custom Filter Setting can be used to include or exclude instances using a filter expression on the name, label, or any other field allowed by GCP. The filter must:

- Be a string, number, or Boolean value
- Use these comparison operators: =, !=, >, or <
- Use parentheses ( ) around each filter
- Combine different filters using AND or OR (all caps). For example: `(name="instanceName") AND (labels.key="value")`

> See Method: instances.aggregatedList for more on filtering instances.

Other useful filters:

Status:

`status="StatusValue"`

`StatusValue` can be `Running` or `Terminated`

Zone:

`zone=https://www.googleapis.com/compute/v1/projects/{ProjectName}/zones/{ZoneName}`

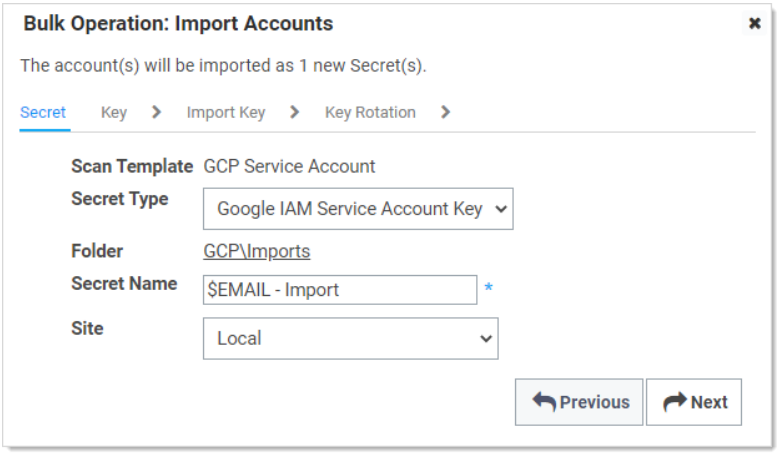
> Unfortunately, at this time of this topic, Google has an open issue of the tag filter not working.

Importing Service Accounts

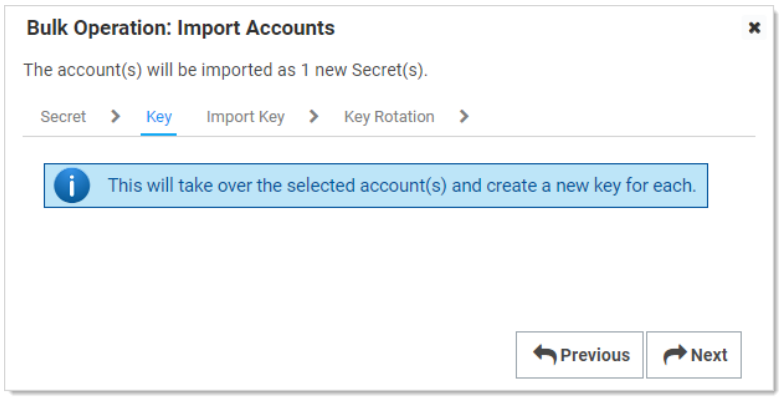
From the Discovery Network View, Secret Server can import Service Account keys and automatically take over the account. This import process will create a new Secret for the Service Account key, delete the associated key, create a new key, and save the json private key file with the Secret, so this can be easily managed by Secret Server.

To Import a Service Account

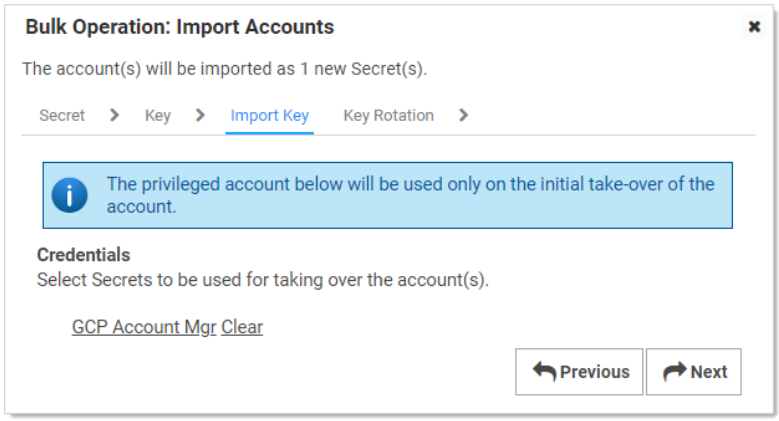
1. Go to **Admin > Discovery**.
2. Click the **Discovery Network View** button. The Discovery Network View page appears.
3. Select the **Domain\Cloud Account** tab
4. Click to select the Service Account(s) to import in the unlabeled Domain/Cloud tree on the left.
5. Click the **Import** button. The importation wizard begins:



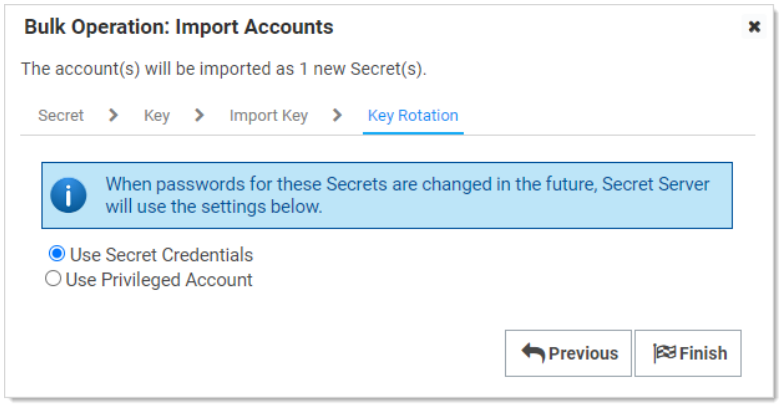
6. For secrets:
   a. Click the **Secret Type** dropdown list and select **Google IAM Service Account Key**.
   b. Click the link after **Folder** to select a folder.
   c. Type a name in the **Secret Name** text box (It auto fills `$EMAIL`).
   d. Click the Site dropdown list to select a site.
7. Click the **Next** button. The Key page appears:



8. When importing GCP service account keys, the only option is take over the account. Meaning, Secret Server triggers a remote password change on import to rotate the imported key and obtain a new JSON private key file. With the JSON private key file, Secret Server can then manage the GCP service account.
9. Click the **Next** button. The Import Key page appears:



10. Click the link to select a secret to use for the initial take over of the account.
11. Click the **Next** button. The Key Rotation page appears:



12. For key rotation, click one of two selection button options to choose a secret for future key rotations. Either option would need the permissions mentioned above. When the password for the chosen secret are changed in the future, Secret Server will use one of these two options:
    - **Use Secret Credentials**: Use the imported service account to rotate itself, and it has permissions to rotate keys.
    - **Use Privileged Account**: Use another service account that has permissions to rotate keys
13. Click the **Finish** button.

GCP APIs

Overview

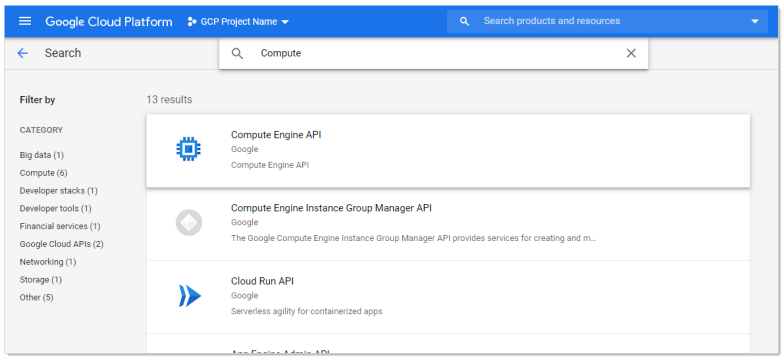
To make API calls to GCP, you need to enable the following APIs to use GCP discovery in Secret Server. More information can be found on the GCP Getting Started page. The APIs are:

- **Cloud Resource Manager API**: Used for managing GCP resource containers, such as Projects.
- **Compute Engine API**: Used for managing GCP instances (virtual machines).
- **Identity and Access Management (IAM) API**: Used for managing identity and access control for GCP resources, such as service accounts.

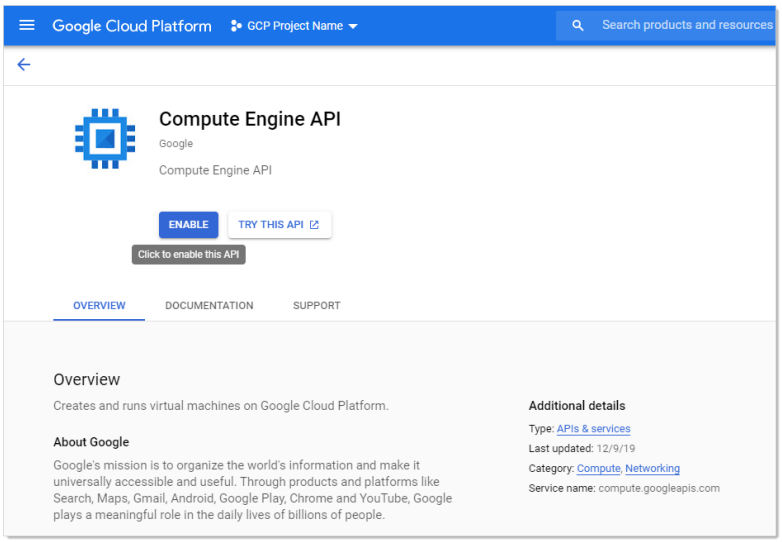
Enabling GCP APIs

In GCP:

1. In GCP, click the **APIs & Services** menu item and select **Library**. The Library page appears.
2. Type the name of the API in the Search text box and press **<Enter>**. Matching APIs appear:



3. Click the button for the desired API. That API's page appears:



4. Click the **Enable** button.

> If you're setting up a new instance and haven't used certain APIs before, you'll need to enable the Identify and Access Management (IAM) API. If you encounter a "Heartbeat Failure" message, just follow the link provided in the message to enable the IAM API. After that, give it a few minutes for the changes to take effect, then try enabling the Compute Engine API again.

Errors and Solutions

Create Keys Failed: Access Denied

Error

```
Create Keys Failed: AccessDenied, Google.Apis.Requests.RequestError Permission
iam.serviceAccountKeys.create is required to perform this operation on service account
projects/-/serviceAccounts/discovery-me@gcpprojectname.iam.gserviceaccount.com. [403] Errors
[ Message[Permission iam.serviceAccountKeys.create is required to perform this operation on
service account projects/-/serviceAccounts/discovery-
me@gcpprojectname.iam.gserviceaccount.com.] Location[ - ] Reason[forbidden] Domain[global] ]
```

Likely Cause

The service account used to rotate the key does not have necessary permission to perform this task.

Solution

1. Go to the GCP console.
2. Select **IAM > Permissions**.
3. Select the service account.
4. Add the **Service Account Key Admin** permission.
5. Once the service account has permission:

a. In Secret Server, select the secret to rotate.

b. Stop the current rotation.

c. Try the operation again.

Create Keys Failed: Maximum Number of Keys on Account Reached

Error

```
Create Keys Failed: ArgumentError, Google.Apis.Requests.RequestError Maximum number of keys
on account reached. [429] Errors [ Message[Maximum number of keys on account reached.]
Location[ - ] Reason[rateLimitExceeded] Domain[global] ]
```

Likely Cause

The rotated service account has reached the maximum number of keys allowed. GCP maximum is 10 keys.

Solution

1. Go to the GCP console.
2. Select **IAM > Permissions**.
3. Remove the unused keys.
4. Once the service account has less than 10 keys, in Secret Server:
   a. In SS, select the secret to rotate.
   b. Stop the current rotation.
   c. Try the operation again.

Discovery Consumer: Syncing OUs Failed

Error

```
DiscoveryConsumer: Synchronizing Organizational Units failed for [Our Google Cloud]! Error:
An issue was encountered during the scan. Google.Apis.Requests.RequestError Access Not
Configured. Compute Engine API has not been used in project 123456 before or it is disabled.
Enable it by visiting
https://console.developers.google.com/apis/api/compute.googleapis.com/overview?project=123456
then retry. If you enabled this API recently, wait a few minutes for the action to propagate
to our systems and retry. [403] Errors [ Message[Access Not Configured. Compute Engine API
has not been used in project 123456 before or it is disabled. Enable it by visiting
https://console.developers.google.com/apis/api/compute.googleapis.com/overview?project=123456
then retry. If you enabled this API recently, wait a few minutes for the action to propagate
to our systems and retry.] Location[ - ] Reason[accessNotConfigured] Domain[usageLimits] ] ,
-2146233088
```

Likely Cause

The discovery service account used for has access to a GCP project that has not been set up or is disabled.

Solution

1. Go to GCP console.
2. Go to **Compute Engine > VM Instances**.
3. Set up the compute engine


This requires billing information.

Discovery Consumer: Syncing Machines Failed

Error

```
DiscoveryConsumer: Synchronizing Machines failed for [GCP Discovery Source]! Error: An issue
was encountered during the scan. Google.Apis.Requests.RequestError Invalid value for field
'filter': 'filtername="value"'. Invalid list filter expression. [400] Errors [ Message
[Invalid value for field 'filter': 'filtername="value"'. Invalid list filter expression.]
Location[ - ] Reason[invalid] Domain[global] ] , -2146233088 Exception Caught:
Google.Apis.Requests.RequestError Invalid value for field 'filter': 'filtername="value"'.
Invalid list filter expression. [400] Errors [ Message[Invalid value for field 'filter':
'filtername="value"'. Invalid list filter expression.] Location[ - ] Reason[invalid] Domain
[global] ] Attempting GCP scan for Instances Parameters are valid. Checking for permissions
to list Projects.. Has permissions to list Projects.. Starting scan..
```

Likely Cause

The instance scanner custom filter is not valid.

Solution

1. In Secret Server, go to the GCP discovery source.
2. Edit the instance scanner.
3. Update the "custom filter" setting.


See Method: instances.aggregatedList for more on filtering instances.

Discovery Consumer: Machine Scan Completed but Computers Failed Authentication

Error

```
DiscoveryConsumer: Synchronizing Machines failed for [GCP Discovery Source]! Error: An issue
was encountered during the scan. Google.Apis.Requests.RequestError Invalid value for field
'filter': 'filtername="value"'. Invalid list filter expression. [400] Errors [ Message
[Invalid value for field 'filter': 'filtername="value"'. Invalid list filter expression.]
Location[ - ] Reason[invalid] Domain[global] ] , -2146233088 Exception Caught:
Google.Apis.Requests.RequestError Invalid value for field 'filter': 'filtername="value"'.
Invalid list filter expression. [400] Errors [ Message[Invalid value for field 'filter':
'filtername="value"'. Invalid list filter expression.] Location[ - ] Reason[invalid] Domain
[global] ] Attempting GCP scan for Instances Parameters are valid. Checking for permissions
to list Projects.. Has permissions to list Projects.. Starting scan..
```

Likely Cause

The instance scanner custom filter is not valid.

Solution

1. In Secret Server, go to the GCP discovery source.
2. Edit the instance scanner.
3. Update the "custom filter" setting.

See Method: instances.aggregatedList for more on filtering instances.

Invalid Grant: Account Not Found

Error

```
An issue was encountered during the scan. Error:"invalid_grant", Description:"Invalid grant:
account not found", Uri:"", -2146233088
```

Likely Cause

The service account does not exist in GCP. There may be a typo or it was deleted.

Solution

1. Go to GCP console.
2. Create a service account to use. See Task 1: Creating GCP Service Accounts.

Request Error: Caller Does Not Have Permission

Error

```
An issue was encountered during the scan. Google.Apis.Requests.RequestError The caller does
not have permission [403] Errors [Message[The caller does not have permission] Location[ - ]
Reason[forbidden] Domain[global]], -2146233088
```

Likely Cause

The service account does not have permissions in IAM.

Solution

1. Go to GCP console.
2. Select IAM.
3. Click the **Service Account** menu item to create a service account with the desire permissions. See Task 1: Creating GCP Service Accounts and Task 2: Setting GCP Permissions.

Unix Account Discovery

Unix account discovery follows these steps:

1. During configuration, Secret Server is given a list of IP address ranges and ports on the network to scan for. See "Creating a Unix Discovery Source" below
2. Within that range, discovery searches for computers listening on the specified ports (default is 22). The ports and other parameters are configurable via the scanners belonging to the discover source. See "Introduction to Discovery Sources, Scanners, and Templates" on page 525
3. Secret Server then attempts to use DNS to resolve the found IPs to discover the associated computer name.
4. Secret Server saves all the collected information to its database.
5. Secret Server then attempts to connect to each computer using the provided credentials and query for a list of user accounts on the target system.

Creating a Unix Discovery Source

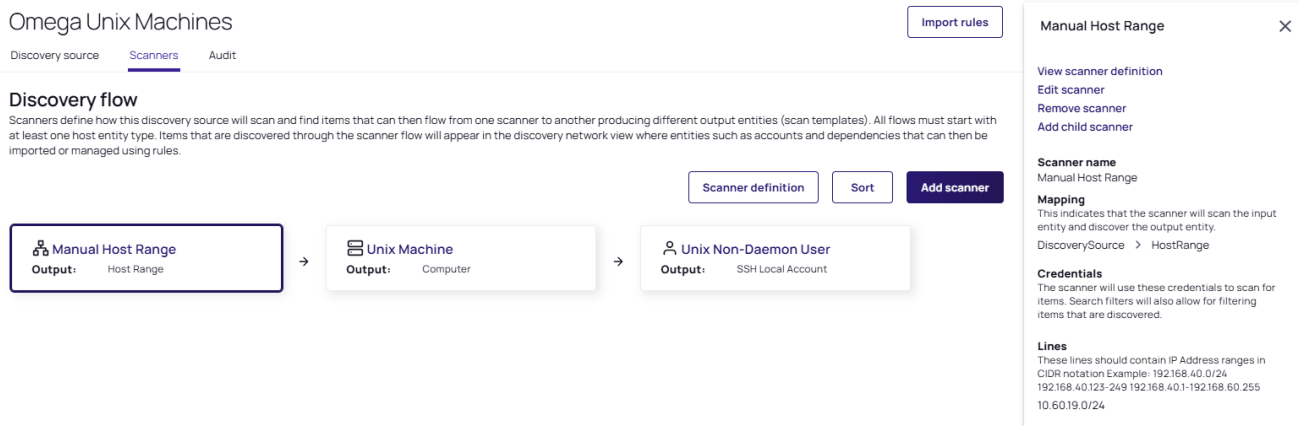
Discovery sources define a set of discovery operations. You must create a discovery source based on the built-in types prior to running discovery.

Creating the Discovery Source

See "Creating a Discovery Source" on page 533 for details.

Editing Unix Discovery Source Scanners

1. Select **Discovery > Sources**. The Sources tab of the Discovery page appears.
2. In the list of existing discovery sources, click the name of the one you want to edit.
3. Select the **Scanners** tab. The Discovery flow for that source and its scanners appear:



> To add a dependency scanner, one needs to be available which has an input template matching an unused output scan template. The output template must be unique for each scanner, but the input template may be shared.

4. **Manual Host Range** is typically the first scanner of the discovery source, which is the input template. This means the initial data comes from information you enter into the discovery source when you create it. The output template is named Host Range.

5. Click the **Manual Host Range** scanner. The details pop-out appears to the right of the screen. Note the following:
   - You have four links at the top, View scanner definition, Edit scanner, Remove scanner, and Add child scanner.
   - The **Mapping** section indicates the scanner will look through the input entity and discover the output entity. For this scanner these are **DiscoverySource** and **HostRange**, respectively.
   - The **Credentials** section displays the secret credentials used to scan for items and allows for filtering.
   - The Unix discovery source finds all machines and local accounts on a set of manually defined host ranges for Unix machines accessible with SSH.
   - The **Lines** section lists multiple, distinct IP address ranges on the same discovery source.
6. The **Unix Machine** scanner is the consumer of the Computer output template, and has the following configurations available that differ from the previous scanner:
   - The **Mapping** section the input entity that is used and what the scan will discover as the output entity. For example: HostRange > Computer.
   - The **Ports** section displays the comma-separated list of port values (1-65535) used for the scanner.
   - The **Max TCP Connections** section determines how many concurrent TCP connections will be used during a scan.
   - The **Attempt Authentication** option allows the server to attempt to login and determine the operating system of any machines found. This requires a Credential Secret and a Command Set.
   - The **Parse Format** option details the format in which the scanner will parse the operating system and hostname from a found machine after the commands are run, when using authenticated scanning.
   - The **Identify computer by host name** option, if true, allows for computers to be considered unique by host name, not just their IP address. This is helpful if one machine is bound to multiple IP addresses. If you change this option after having run discovery any secrets previously linked will not automatically transfer to the new computer.
7. The **Unix Non-Daemon User** scanner has the following configurations available that differ from the previous scanners:
   - The **User Regex Format** text box contains a regular expression that finds the lines of text received during the scan that are valid for user parsing. The matched groups in the regular expression should correspond to the comma-separated items in the parse format.
   - The **Parse Format** text box defines the order of values retrieved during a scan. If the parse names match the fields defined in the imported secret, the values are populated from the data collected on the scan.
   - The **Newline Separator Character** dropdown defines the character that divides the lines in the output received during a scan.

There are other discovery source-specific scanners available for use with different options to choose from. Each option is detailed when the scanner is selected.

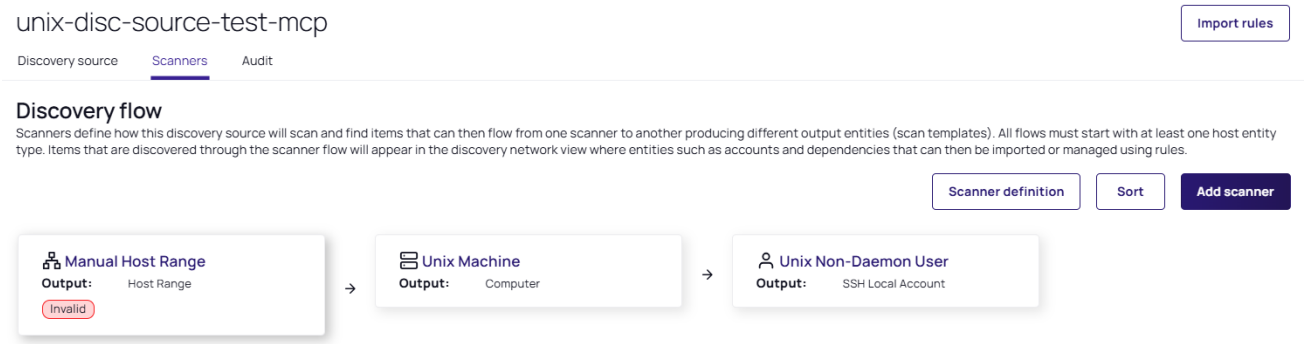
Discovering SSH Public Keys

Secret Server can scan for SSH public keys on Unix machines. You can add this ability in the scanner settings section of Unix Account Discovery.

> This document assumes you have already created a Unix account discovery source. See "Creating a Unix Discovery Source" on page 634.

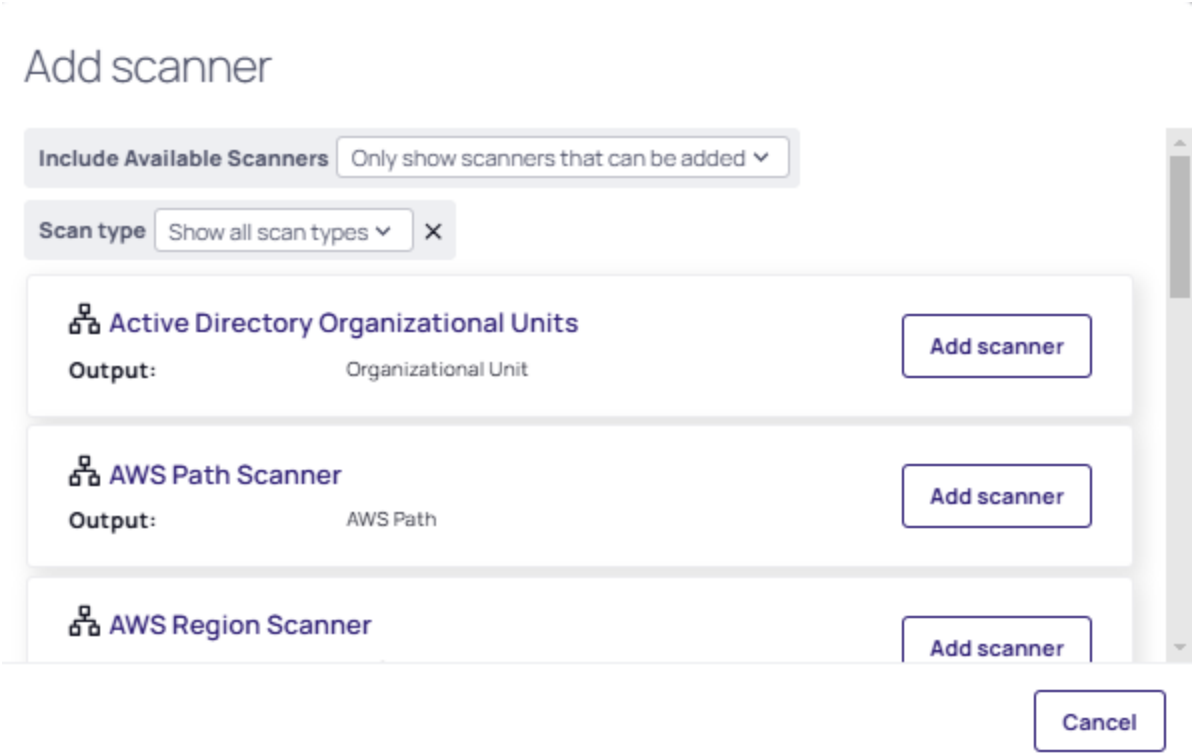
Task 1: Viewing Discovery Scanners for the Unix Discovery Source

1. Access **Discovery** from the side menu and select **Sources**.
2. Select the Unix discovery source that you created in the previous section. The **Discovery Source** tab for it appears.
3. Select the **Scanners** tab. The discovery source **Scanners** page appears:



Task 2: Adding the SSH Public Key Scanner

1. In the discovery source **Scanners** page, click **Add Scanner**. The Add Scanner popup appears:



2. Select the **SSH Public Key Scanner**.
3. Select the newly added **SSH Public Key Scanner**. The details open in a panel on the right side of the screen.
4. Click **Edit Scanner**. Various sections in the details panel are now editable.
5. Click the **Add Secret** link and choose one or more secrets that have Unix sudo or su permissions for the host range selected in the discovery source.

   These permissions are necessary to navigate each user's home directory on a machine in search of SSH public key entries in the user's `<user home directory>/.ssh/authorized_keys` file.
6. Click **Save**.

Task 3: Importing SSH Public Keys

From the **Discovery Network View**, Secret Server can import SSH public keys and potentially take over an account. The import process creates a new secret for the SSH public key in one of two ways:

- Including a provided matching SSH private key and passphrase.
- Taking over the key by creating a new key and saving the private key file and passphrase with the secret. This can be easily managed by Secret Server.

Prerequisites

Before proceeding with importing an SSH Public Key based account, ensure that an SSH key pair has been generated and that the public key is available for import.

If you do not already have an SSH key for your account user, follow these steps:

1. Generate one using the following command:

   `ssh-keygen -t rsa -b 4096 -C "your-email@example.com"`

   This creates a public key (`id_rsa.pub`) and a private key (`id_rsa`) in the `~/.ssh/ directory` (Linux/macOS) or `C:\Users\YourUser\.ssh\` (Windows).

   
   The public key (`id_rsa.pub`) is the file that will be imported.

2. If you already have an SSH key pair, verify that the public key exists by running:

   `cat ~/.ssh/id_rsa.pub`

   If no output appears, an SSH key may not have been created or may be stored under a different filename.

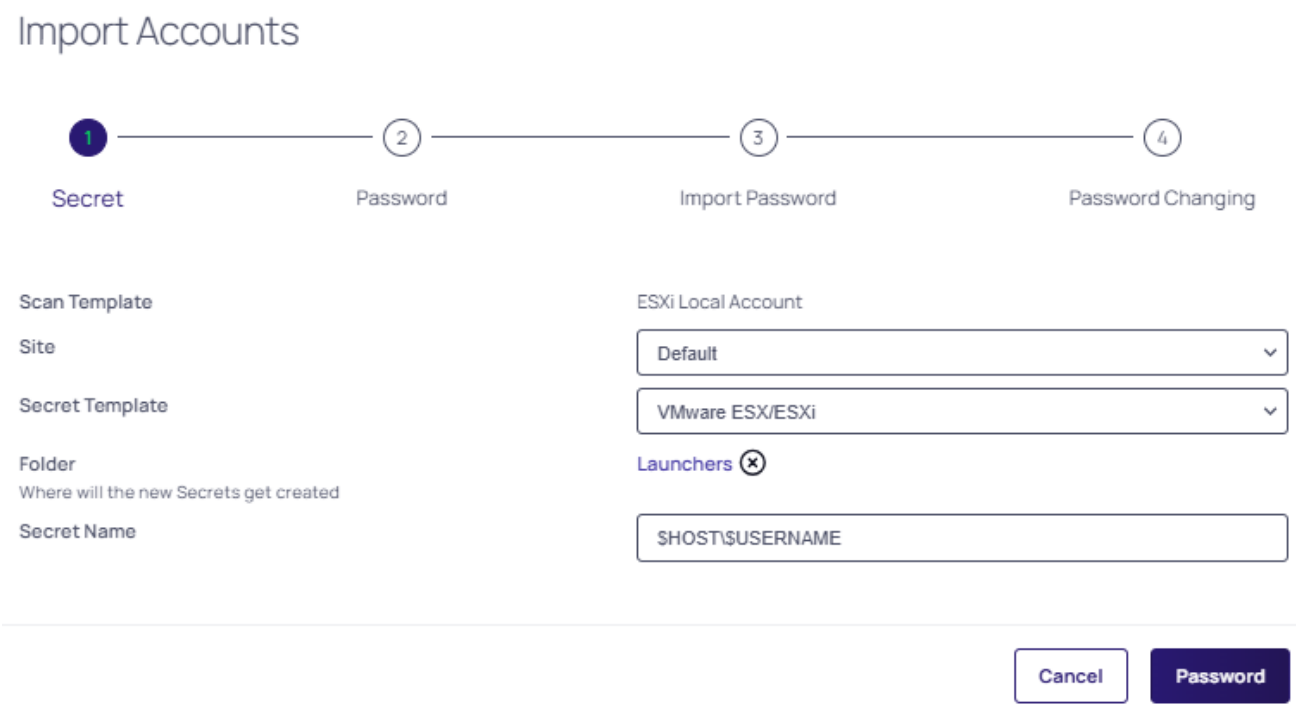
3. Before importing, verify whether the public key must be linked to an account in the system:
   - If an account must have a public key before import, manually add the key:

     `ssh-copy-id username@remote-server`
   - If the import process automatically associates the key, ensure the system can discover existing public keys.
   - If accessing a remote server, ensure the key is added to the `~/.ssh/authorized_keys` file on that server.

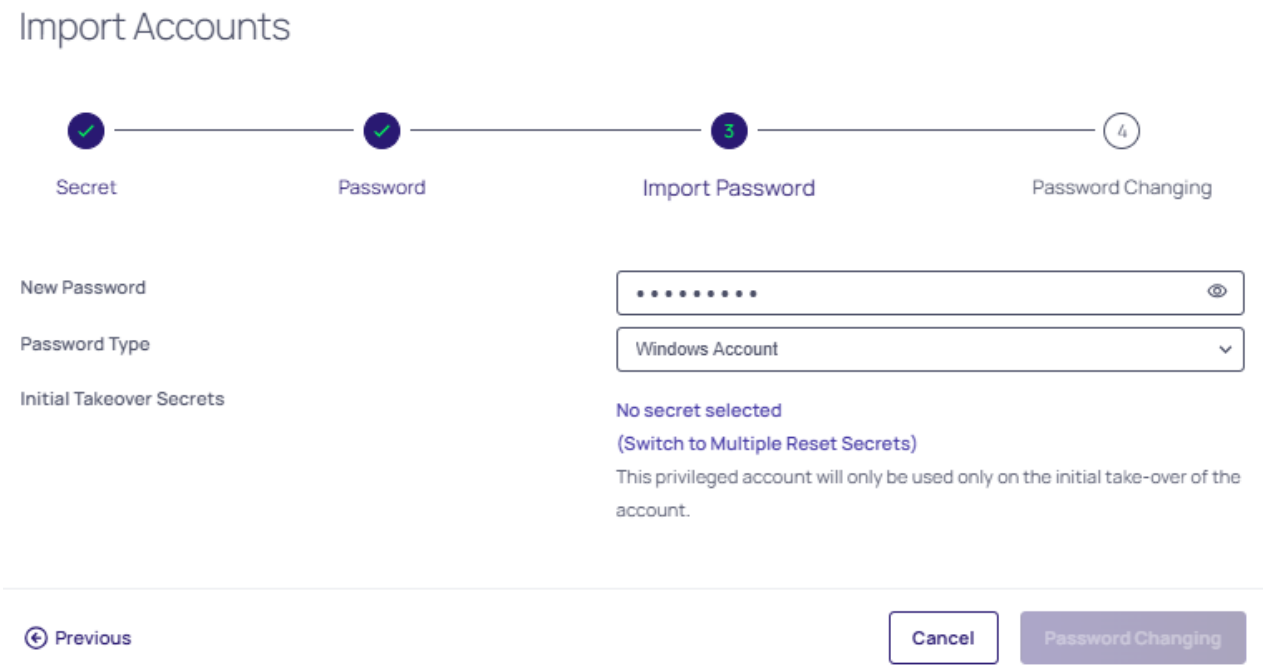
   Once the public key is available, proceed with the steps outlined in "To Import an SSH Public Key" below.

To Import an SSH Public Key

1. Access **Discovery > Network View**. The Discovery Network View page appears listing all discovered and managed items.
2. Set the **Item type** option to **Computer accounts**.
3. Select the account with your associated key from the list. The Computer account detail panel opens to the right.
4. Click the **Actions** dropdown list and choose **Import**. The import Accounts popup appears:

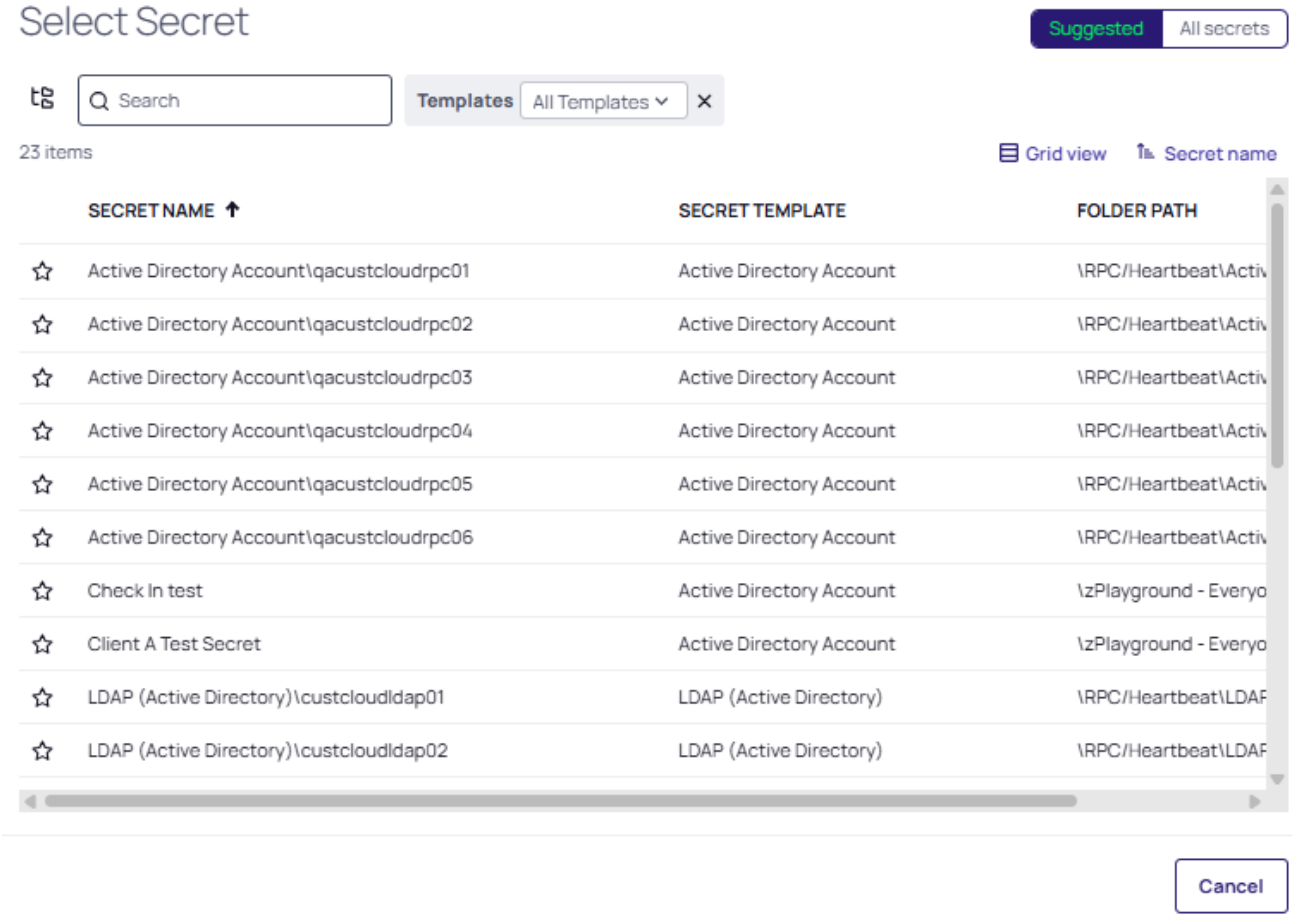


5. Click the **Site** dropdown list to select a site.
6. Click the **Secret Template** dropdown list to select a template. Usually only one is available.
7. Click the **Folder** link to select a folder. This is where will the new secrets will be created.
8. Type a name in the **Secret Name** text box. (It auto fills with `$MACHINE\$USERNAME`).
9. Click the **Password** button. The Password screen in the popup appears.
10. Select one of the three options:
    - **I know the current password and do not want to change it**
    - **Assign a new specific password to all accounts**
    - **Generate a random password for each account**
11. Click the **Import Password** button, the following will happen based on your choice:
    - If you chose "I know the current password and do not want to change it" you will be prompted to provide the current password. before being able to click **Password Changing**.
    - If you choose "Assign a new specific password to all accounts" you will be prompted to type in a new password, choose the password type, and choose the Initial Takeover Secrets from the available link:

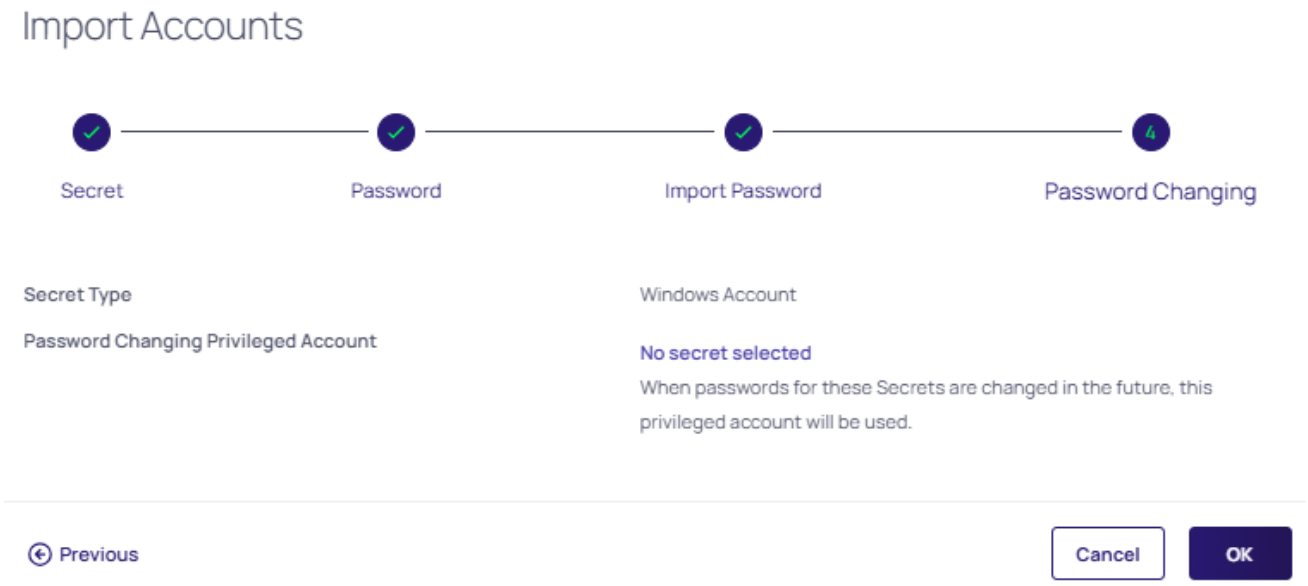


- If you chose "Generate a random password for each account" you will be prompted to choose the password type and the Initial Takeover Secrets from the available link.

12. For the two choices that require selecting Initial Takeover Secrets from the available link, choose a secret:



13. Click the **Password Changing** button, the Password Changing section appears:



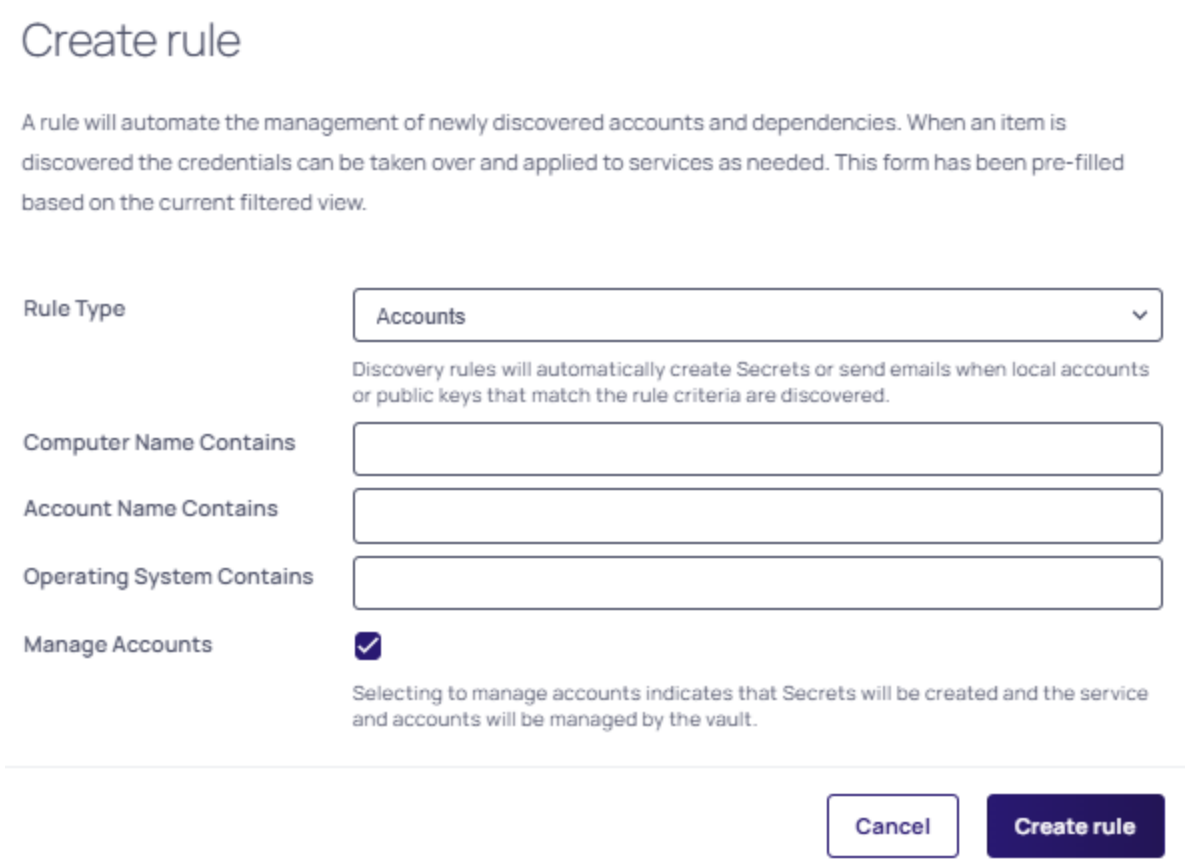
14. Click the **Add Secret** link to choose a Unix sudo or su secret for future key rotations.
15. Click the **OK** button to complete the dialog and import the selected secrets.
16. The **Bulk Progress** popup will appear as the process runs. If successful, you will see "Task complete: DiscoveryImport".
17. Click **Close** when done. The new secret will now appear in the folder you designated previously.

Task 4: Creating an Import Rule

Discovery rules automatically create secrets and send emails when local accounts or public keys match the rule.

To create a rule to import discovered SSH public keys:

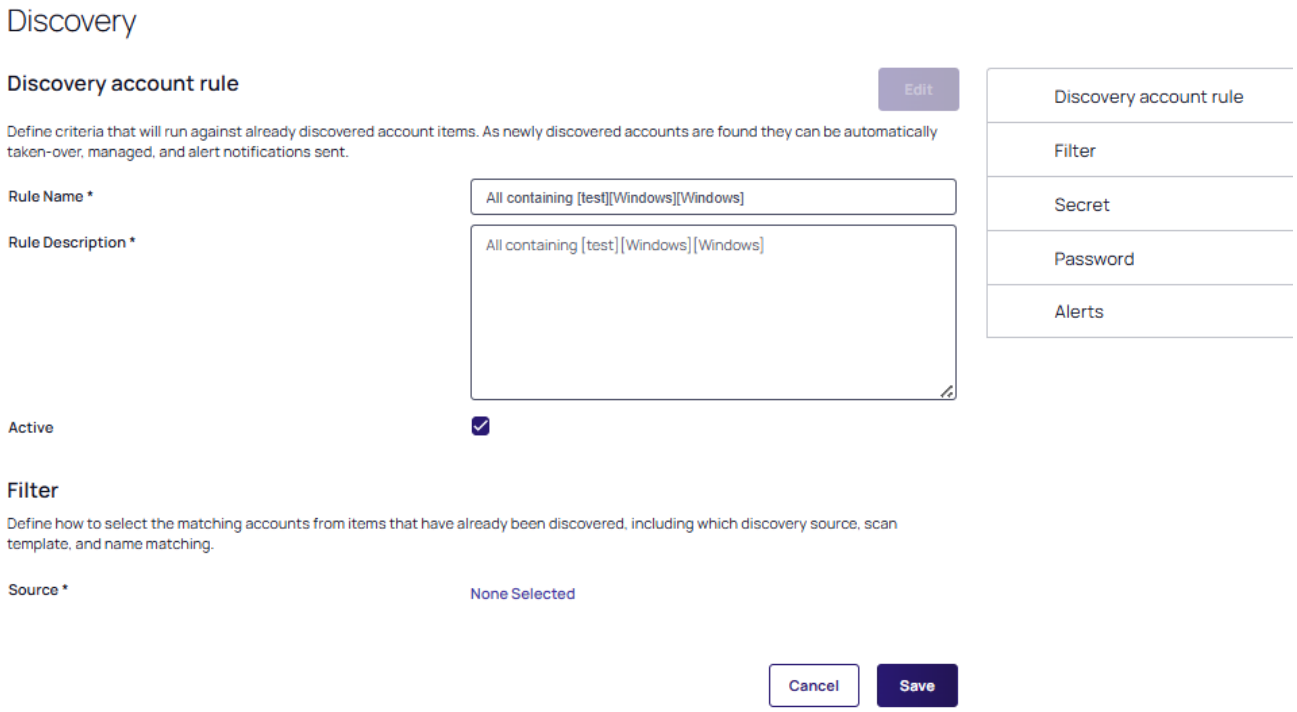
1. Access **Discovery > Network View**. The Discovery Network View page appears.
2. Click the **Create Rule** button. The Create Rule wizard appears:



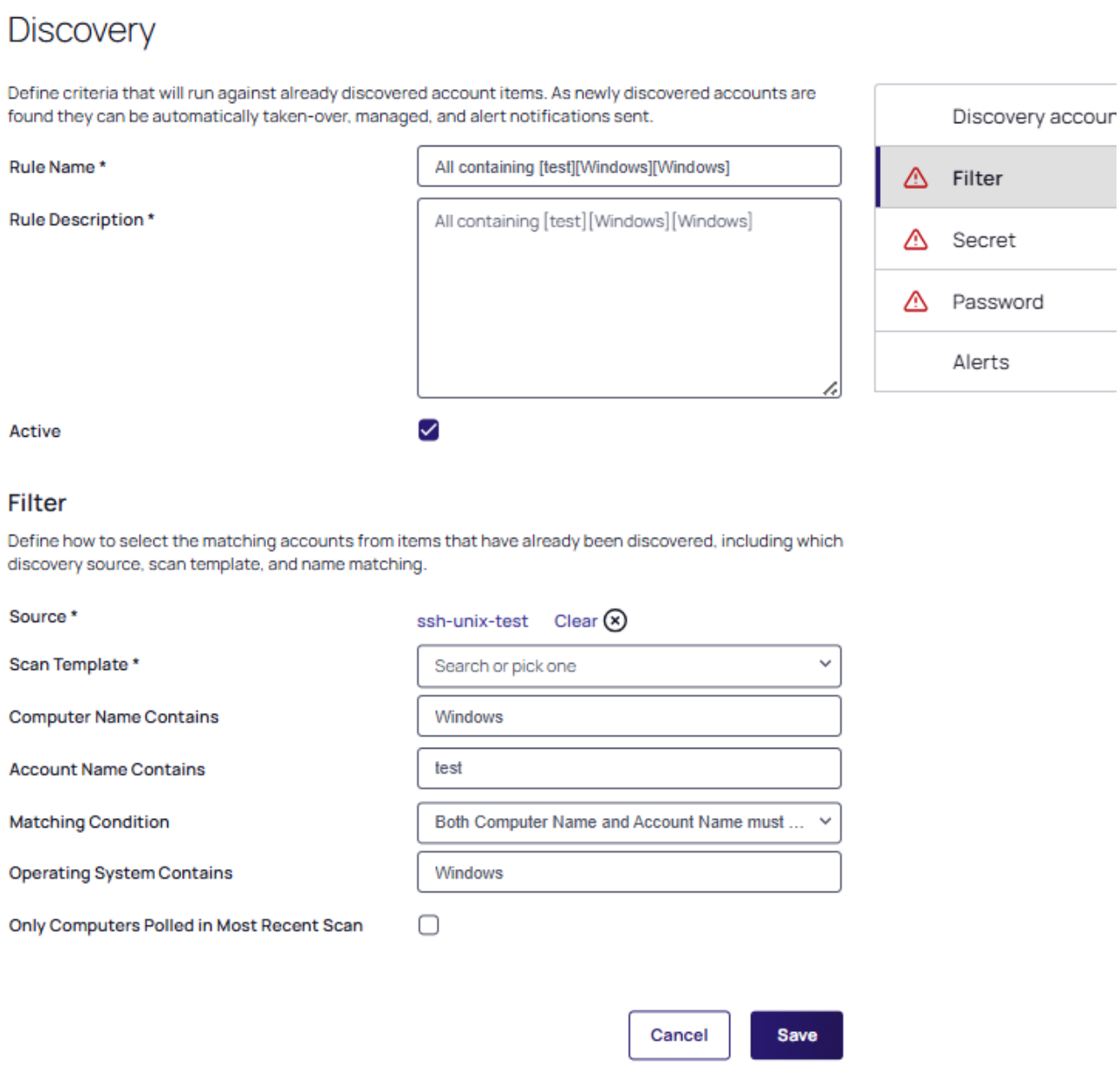
3. Select the **Rule Type**: Accounts or Dependencies.
4. Fill in the following fields if you chose Accounts:
   - Computer Name Contains
   - Account Name Contains
   - Operating System Contains

> Make sure the Manage Accounts checkbox is selected. Selecting to manage accounts indicates that secrets will be created and the service and accounts will be managed by the vault.

5. Click **Create Rule** to save your changes. The Discovery Account Rule page appears:



6. The **Rule Name** field contains the values of the three fields you completed in step 4. The Rule Description contains a duplicate of the rule name values. You can change both manually.
7. Make sure the **Active** checkbox is selected.
8. Under the **Filter** section, click on the **Source** link that displays as **None Selected**. The Pick OU popup appears.
9. Select the Unix source you created in step 1 from the list and the Filter section will automatically display fields like the following:



10. Most of the fields are completed automatically. Choose the **Scan Template** from the dropdown list, in this case **SSH public key**:

Filter

Define how to select the matching accounts from items that have already been discovered, including which discovery source, scan template, and name matching.

Source *: ssh-unix-test Clear

Scan Template *: SSH public key

Computer Name Contains: Windows

Account Name Contains: test

Matching Condition: Both Computer Name and Account Name must ...

Operating System Contains: Windows

Only Computers Polled in Most Recent Scan: ☐

Public key:

Secret

Define where, how, and what type of new Secret will be created

Create Secrets: ☑

Secret Template *: Search or pick one

Folder *: No Folder Selected

Secret Name *: $MACHINE$USERNAME

New Secret Permissions *: Copy permissions from folder

Site *: Search or pick one

Alerts

Send notifications of newly discovered accounts or threshhold limits

Accounts Found Notification: ☐ Send email alert for newly discovered accounts

Cancel Save

11. You can alter the **Marching Condition** and select or deselect the **Only Computers Polled in Most Recent Scan** checkbox.
12. Add in the Public Key for your account user (previously created or associated in the prerequisites).
13. In the **Secret** section under **Filter** define where, how, and what type of new secret will be created:
    a. Select the **Create Secrets** checkbox.
    b. From the **Secret Template** dropdown list, choose one of the two options: **Unix Account (SSH Key Rotation - No Password)** or **Unix Account (Privileged Account SSH Key Rotation - No Password)**.
14. Click the **Folder** link to select a folder in which the secret will be created.
15. Type a name in the **Secret Name** text box (It auto fills `$MACHINE\$USERNAME Key`).
16. Click the **New Secret Permissions** dropdown list to choose how permissions are propagated for the new secret, either copied or inherited from the folder.
17. Click the **Site** dropdown list to select a site. The **Password** section appears.
18. In the **Password** section, select one of the following options:
    a. **I have the matching private key** - This choice causes the following to appear:
        i. **Current Private Key field**: copy and paste the account's current private key in this field.
        ii. **Current Passphrase**: copy and paste the account's current passphrase in this field.
        iii. **Password Changing Privileged Account**: click the link to select a secret.

        > When passwords for the secret added here are changed in the future, this privileged account will be used.

    b. **I want to change the public SSH key on the Account** - This choice causes the following to appear:
        i. **Take-over threshhold**: this field displays 1000 by default and can be changed to another value.

        > If the number of accounts that will be taken over exceeds the max threshold, the import is canceled and the subscribed users below are notified by email.

        ii. **Initial Takeover Secrets**: this button allows you to add multiple secrets to modify.

        This privileged account will only be used on the initial takeover of the account.

        iii. **Password Changing Privileged Account**: this button allows you to add multiple secrets to modify.

        When passwords for these secrets are changed in the future, this privileged account will be used.
19. In the **Alerts** section, you have the option of selecting to send an email alert for newly discovered accounts.

    Example:

Secret

Define where, how, and what type of new Secret will be created

Create Secrets ☑

Secret Template * Unix Account (Privileged Account SSH Key Rot...

Folder * Miruna Paun ⊗

Secret Name * $MACHINE\$USERNAME Key

New Secret Permissions * Copy permissions from folder

Site * GAMMA

Password

Specify how to gain access to this account and whether or not it should be managed

Password *
- ○ I have the matching private key.
- ◉ I want to change the public SSH key on the Account.

Take-over threshhold * 1000

If the number of accounts that will be taken over exceeds the max threshold, the import is cancelled and the subscribed users below are notified by email.

Initial Takeover Secrets * Add Secret

- Check In test Remove

This privileged account will only be used only on the initial take-over of the account.

Password Changing Privileged Account Add Secret

- ad-disc-test-mcp Remove

When passwords for these Secrets are changed in the future, this privileged account will be used.

Alerts

Send notifications of newly discovered accounts or threshhold limits

Accounts Found Notification ☐ Send email alert for newly discovered accounts

Cancel Save

20. Click **Save** when all your settings are complete.
21. Return to the **Discovery Network View** page.
22. Click **View Rules** and select the **Account** or the **Dependency** tab depending on your rule type, to see a list of rules including the one you just created.

Troubleshooting SSH Key Import Issues

If the import fails, consider the following:

- **Public key not found**: Ensure the key exists in `~/.ssh/` and is readable.
- **Authentication failures**: Check that the public key is correctly associated with the target account.
- **Debugging SSH authentication**: Increase logging verbosity with by using the `ssh -vvv username@remote-server` command.
- **Scanning for public keys**: If the system supports key discovery, verify that the process is correctly configured.

VMware ESX/ESXi Account Discovery

During configuration, Secret Server is given a list of IP addresses or computer names that correspond to ESX or ESXi servers. Secret Server then connects to each server using the provided credentials to query for a list of user accounts on the target system.

VMware ESX/ESXi Account Discovery and RPC Configuration

Please see Discovery General Topics for a comprehensive guide to configuring and using discovery.

Download Locations

- Run the following PowerShell command:

  ```
  Install-Module -Name VMware.PowerCLI -Force
  ```

- If the server does not have internet access, download supported versions of PowerCLI from VMware using the following link: VMware PowerCLI

You should use this link only if the server does not have internet access. The preferred installation method is using the PowerShell command above.

Overview

The ESX/ESXi (API) password changer verifies (using heartbeat) and changes VMware ESX/ESXi passwords via the vSphere API. Password changing and discovery for Secret Server 10.6 and later requires PowerCLI 6.5.1 or higher.

Either PowerCLI 6.5.1 or higher must be installed on the servers running discovery or your local Secret Server machine or machines running distributed engine. Earlier versions of the password changer are now deprecated.

If you get an error about not being able to load file or assembly VMware.Binding.Wcf.dll or VMware.Binding.WsTrust.dll, you can copy either file from location: *C:\Program Files\WindowsPowerShell\Modules\VMware.VimAutomation.Common\X.X.X.X\net45* - where *X.X.X.X* is the number of version installed on your system, and paste it to the location: *C:\Program Files\WindowsPowerShell\Modules\VMware.Vim\X.X.X.X\net45* - where *X.X.X.X* is the number of version installed on your system.

Starting with version 13.1.0 the VMware PowerCLI installation files no longer have *net45* folders. For versions 13.1.0 and greater use the *net472* folders instead.

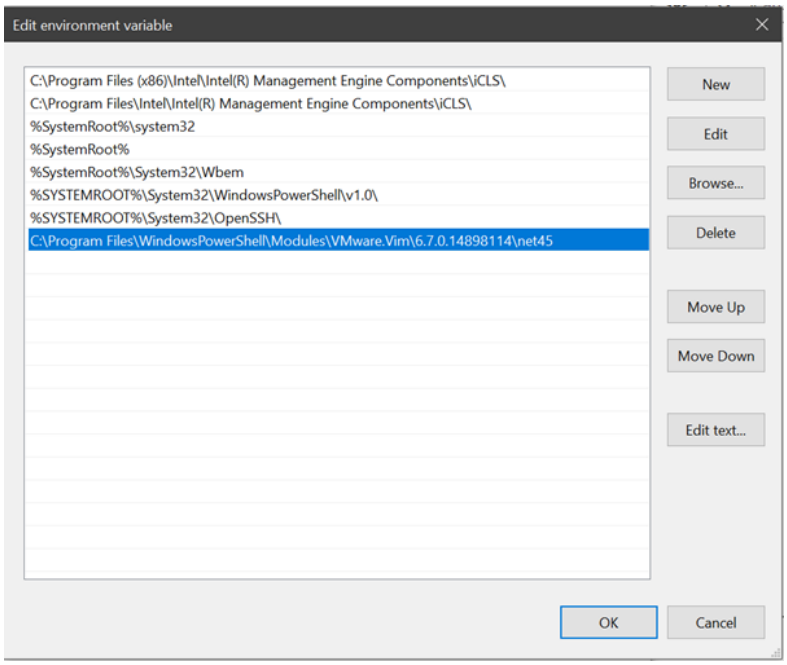
Details

Secret Server searches the machine's Windows PATH environment variable for the VMware SDK, therefore installing the correct version of it is all that is needed. On the machine you install VMware PowerCLI, update the Windows PATH environment variable to include the folder where the file `VMware.Vim.dll` is located.

After installing the VMware PowerCLI, the default installation path is: `C:\Program Files\WindowsPowerShell\Modules\VMware.Vim\[version]\net45`. The PowerCLI installation path **must be** in the system PATH variable.

To edit your PATH:

1. Add `C:\Program Files\WindowsPowerShell\Modules\VMware.Vim\[version]\net45` to the PATH using the system panel (sysdm.cpl).
2. From the **System Properties** dialog, select **Advanced** tab
3. Click **Environment Variables...**
4. Under the **System Variables** section, highlight **Path** then **Edit**. The Edit Environment Variable dialog box appears:



5. Click**New**.
6. Type `C:\Program Files\WindowsPowerShell\Modules\VMware.Vim\[version]\net45`, similar to the example above.
7. Click**OK**when done.

> Note that starting with version 13.1.0 the VMware PowerCLI installation files no longer have *net45* folders. For versions 13.1.0 and greater use the *net472* folders instead.

Setting Up and Troubleshooting VMWare ESXi with Secret Server

Run the following script as an Administrator:

```
#region Helper Functions

function Ensure-NuGetProvider {
    if (-not (Get-PackageProvider -Name NuGet -ErrorAction SilentlyContinue)) {
        Write-Host "NuGet provider is not installed. Installing..."
        Install-PackageProvider -Name NuGet -MinimumVersion "2.8.5.201" -Force -Scope
CurrentUser
    }
}

function Get-ModuleFolderPath {
    param (
        [Parameter(Mandatory)]
        [string]$ModuleName
    )
```

```
    $mod = Get-Module -ListAvailable -Name $ModuleName | Sort-Object Version -Descending |
Select-Object -First 1
    if ($mod) { return $mod.ModuleBase } else { return $null }
}

function Get-NetFolderPath {
    param (
        [Parameter(Mandatory)]
        [string]$ModuleFolder
    )
    foreach ($net in @("net472", "net45")) {
        $path = Join-Path -Path $ModuleFolder -ChildPath $net
        if (Test-Path $path) { return $path }
    }
    return $null
}

function Restart-ThycoticService {
    param(
        [string]$ServiceName = "Thycotic.DistributedEngine.Service",
        [int]$WaitIntervalSeconds = 15,
        [int]$MaxWaitSeconds = 60
    )
    Write-Host "Restarting $ServiceName..."
    $null = Stop-Service -Name $ServiceName -Force -ErrorAction SilentlyContinue -
WarningAction SilentlyContinue
    Write-Warning ("Service '{0}' is stopping. Waiting up to {1} seconds..." -f
$ServiceName, $MaxWaitSeconds)
    $elapsed = 0
    while ((Get-Service -Name $ServiceName).Status -ne "Stopped" -and ($elapsed -lt
$MaxWaitSeconds)) {
        Start-Sleep -Seconds $WaitIntervalSeconds
        $elapsed += $WaitIntervalSeconds
    }
    try {
        Start-Service -Name $ServiceName
        Write-Host "$ServiceName restarted successfully."
    }
    catch {
        Write-Error ("Failed to restart {0}: {1}" -f $ServiceName, $_.Exception.Message)
    }
}

function Add-ToSystemPath {
    param(
        [Parameter(Mandatory)]
        [string]$PathToAdd
    )
    try {
        $currentPath = [System.Environment]::GetEnvironmentVariable("PATH",
[System.EnvironmentVariableTarget]::Machine)
        if ($currentPath -notlike "*$PathToAdd*") {
            $newPath = $currentPath + ";" + $PathToAdd
```

```
            [System.Environment]::SetEnvironmentVariable("PATH", $newPath,
[System.EnvironmentVariableTarget]::Machine)
            Write-Host ("Added '{0}' to the system PATH variable. A shell restart may be
required." -f $PathToAdd)
        }
        else {
            Write-Host ("'{0}' is already in the system PATH." -f $PathToAdd)
        }
    }
    catch {
        Write-Warning ("Failed to update system PATH: {0}" -f $_.Exception.Message)
    }
}

function Install-PowerCLIAndCopyFiles {
    Write-Host "=== Installing PowerCLI and copying required files for Secret Server ==="
-ForegroundColor Cyan

    Ensure-NuGetProvider
    if (-not (Get-Module -ListAvailable -Name VMware.PowerCLI)) {
        Write-Host "PowerCLI not found. Installing VMware.PowerCLI..."
        try {
            Install-Module -Name VMware.PowerCLI -Force -Scope CurrentUser -AllowClobber
        }
        catch {
            Write-Error ("Failed to install VMware.PowerCLI: {0}" -f $_.Exception.Message)
            return
        }
    }
    else {
        Write-Host "PowerCLI is already installed."
    }
    $commonModuleBase = Get-ModuleFolderPath -ModuleName "VMware.VimAutomation.Common"
    $vimModuleBase    = Get-ModuleFolderPath -ModuleName "VMware.Vim"
    if (-not $commonModuleBase) {
        Write-Error "Could not locate the VMware.VimAutomation.Common module folder."
        return
    }
    if (-not $vimModuleBase) {
        Write-Error "Could not locate the VMware.Vim module folder."
        return
    }
    $sourcePath = Get-NetFolderPath -ModuleFolder $commonModuleBase
    $destinationPath = Get-NetFolderPath -ModuleFolder $vimModuleBase
    if (-not $sourcePath) {
        Write-Error ("Source directory (net472/net45) not found under {0}." -f
$commonModuleBase)
        return
    }
    if (-not $destinationPath) {
        Write-Error ("Destination directory (net472/net45) not found under {0}." -f
$vimModuleBase)
        return
    }
```

```
    $files = @("VMware.Binding.Wcf.dll", "VMware.Binding.WsTrust.dll")
    foreach ($file in $files) {
        $sourceFile = Join-Path -Path $sourcePath -ChildPath $file
        $destFile = Join-Path -Path $destinationPath -ChildPath $file
        if (Test-Path $sourceFile) {
            Write-Host ("Copying {0} from {1} to {2}..." -f $file, $sourcePath,
$destinationPath)
            try {
                Copy-Item -Path $sourceFile -Destination $destFile -Force
                Write-Host ("Copied {0} successfully." -f $file)
            }
            catch {
                Write-Warning ("Failed to copy {0}: {1}" -f $file, $_.Exception.Message)
            }
        }
        else {
            Write-Warning ("File {0} was not found in {1}." -f $file, $sourcePath)
        }
    }
    Write-Host "Updating the system PATH variable..."
    Add-ToSystemPath -PathToAdd $destinationPath
    $restart = Read-Host "Do you want to restart Thycotic.DistributedEngine.Service now?
(Y/N)"
    if ($restart -match "^[Yy]") {
        Restart-ThycoticService
    }
    else {
        Write-Host "Reminder: Restart the Thycotic.DistributedEngine.Service later for
changes to take effect."
    }
}

#endregion Helper Functions

function Test-VMWareHeartbeat {
    param (
        [string]$esxiHost,
        [PSCredential]$cred
    )
    Write-Host "=== Testing VMware ESXi Heartbeat ===" -ForegroundColor Cyan
    Write-Host ("Attempting to connect to {0} with default cert policy (Unset)..." -f
$esxiHost)
    Write-Host "This could take up to 5 minutes (connection will time out after 5 mins)"

    Set-PowerCLIConfiguration -Scope Session -InvalidCertificateAction Unset -
Confirm:$false | Out-Null
    Set-PowerCLIConfiguration -Scope Session -DefaultVIServerMode Single -Confirm:$false |
Out-Null

    try {
        Connect-VIServer -Server $esxiHost -Credential $cred -ErrorAction Stop | Out-Null
        Write-Host "Connection succeeded with default cert policy." -ForegroundColor Green
        Disconnect-VIServer -Server $esxiHost -Confirm:$false | Out-Null
```

```
        return
    }
    catch {
        Write-Warning "Default connection failed due to certificate issues."
    }

    Write-Host "Setting session cert policy to Ignore (session only) and retrying..." -
ForegroundColor Yellow
    Set-PowerCLIConfiguration -Scope Session -InvalidCertificateAction Ignore -
Confirm:$false | Out-Null
    Set-PowerCLIConfiguration -Scope Session -DefaultVIServerMode Single -Confirm:$false |
Out-Null

    try {
        Connect-VIServer -Server $esxiHost -Credential $cred -ErrorAction Stop | Out-Null
        Write-Host "Connection succeeded with certificates ignored (session only)." -
ForegroundColor Green
    }
    catch {
        Write-Error "Heartbeat failed even with certificates ignored: $_"
    }
    finally {
        Disconnect-VIServer -Server $esxiHost -Confirm:$false | Out-Null
    }
}

function Test-VMWareDiscovery {
    param (
        [string]$esxiHost,
        [PSCredential]$cred
    )
    Write-Host "=== Testing VMware Host Account Discovery ===" -ForegroundColor Cyan
    Write-Host "Attempting to connect to $esxiHost with default cert policy (Unset)..."
    Write-Host "This could take up to 5 minutes (connection will time out after 5 mins)"

    # Phase 1: default Unset
    Set-PowerCLIConfiguration -Scope Session -InvalidCertificateAction Unset -
Confirm:$false | Out-Null
    try {
        $connection = Connect-VIServer -Server $esxiHost -Credential $cred -ErrorAction
Stop
    }
    catch {
        Write-Warning "Default connection failed due to certificate issues."
        Write-Host "Setting session cert policy to Ignore (session only) and retrying..."
-ForegroundColor Yellow
        Set-PowerCLIConfiguration -Scope Session -InvalidCertificateAction Ignore -
Confirm:$false | Out-Null
        $connection = Connect-VIServer -Server $esxiHost -Credential $cred -ErrorAction
Stop
    }

    Write-Host "Connection successful. Scanning for user accounts on $esxiHost..."
```

```
    # Retrieve all accounts via the connected server
    $accounts = Get-VMHostAccount -Server $connection -ErrorAction Stop
    if ($accounts) {
        $accounts | Format-Table -AutoSize
    }
    else {
        Write-Host "No user accounts were found." -ForegroundColor Yellow
    }

    Disconnect-VIServer -Server $connection -Confirm:$false | Out-Null
}

#region Main Script Prompt

Write-Host ""
Write-Host "Select an option:" -ForegroundColor Green
Write-Host "1. Install PowerCLI and copy required files for Secret Server"
Write-Host "2. Test a VMware ESXi Heartbeat"
Write-Host "3. Test scanning a VMware host for accounts (Discovery)"
$choice = Read-Host "Enter your choice (1, 2, or 3)"

switch ($choice) {
    "1" { Install-PowerCLIAndCopyFiles }
    "2" {
        $esxiHost = Read-Host "Enter the ESXi host name or IP for Heartbeat test"
        $cred = Get-Credential -Message "Enter credentials for connecting to the ESXi
host"
        Test-VMWareHeartbeat -esxiHost $esxiHost -cred $cred
    }
    "3" {
        $esxiHost = Read-Host "Enter the ESXi host name or IP for Discovery scan"
        $cred = Get-Credential -Message "Enter credentials for connecting to the ESXi
host"
        Test-VMWareDiscovery -esxiHost $esxiHost -cred $cred
    }
    Default { Write-Host "Invalid selection. Exiting." }
}

#endregion Main Script Prompt
```

Troubleshooting and Issues

- The error "The VMware VIM API is not installed or is the wrong version" indicates that PowerCLI needs to be installed.
- We recommend not using an outdated SDK with an updated version of VMware.
- Secret Server's VMware password changer rejects self-signed SSL certificates. Make sure your VMware servers have valid SSL certificates (see below for settings).
- The error "Exception: The remote certificate is invalid according to the validation procedure" indicates that vCenter server root certificates needs to be installed. For more information, see Download and install vCenter

Server root certificates to avoid web browser certificate warnings.

- For Secret Server installed editions, you may need to restart the Secret Server website after installing PowerCLI. Do this by recycling the Secret Server application pool or performing an IIS reset.
- For distributed engines, the distributed engine service may need to be restarted after PowerCLI is installed.

ESXi Certificate Settings

> VMware recommends not including a CRL/CDP in certificate templates. To that end, we recommend adding the `X509RevocationMode.NoCheck` option to the `ESXi.CertificateChainPolicyOptions` setting.

Delinea added a configuration option for Secret Server to allow ESXi TLS connections to ignore self-signed certificates, allow certificates from specific issuers (even if issuer is not in trusted certificate lists), or completely skip certificate validation when using ESXi password changer, heartbeat, or discovery.

> For security reasons, we do **not** encourage customers to use self-signed certificates. Therefore, the new configuration settings listed below are not accessible through the UI. If you need to alter the default ESXi certificate validation settings, **submit a case through Delinea's Support Portal** for assistance.

New advanced configuration settings include:

- ESXi: Enable TLS Debugging and Connection Tracking: Identical to TLS Audit option, but specifically for ESXi. If set to true and Secret Server (or DE) auditing is set to DEBUG, detailed debug messages about the certificate chain will be written to the log file.
- ESXi: Certificate Chain Policy Options: Identical to TLS Audit option, but specifically for ESXi. Allows setting X509 options to be applied to certificate validation. This is a comma-delimited list of options. See TLS Auditing or the Details section for more information.
- ESXi: Allow Certificates Issued By: Semi-colon delimited list of issuer names (in format shown on certificate---such as "O=Issuer Name"). Ignores partial chain errors due to certificate being issued by any issuer in this list when that issuer is not in the trusted root or intermediate CAs lists on the server.
- ESXi: Client Certificate Thumbprints: Identical to TLS Audit option, but specifically for ESXi. If ESXi host requires the client to present a valid certificate, this is a semi-colon delimited list of client certificates on the server to try to present.
- ESXi: Ignore All Certificate Errors: If true, certificate validation will not be performed. All certificate errors will be ignored.
- ESXi: Allow Default Host Certificates: Sets all the TLS configuration options necessary to not fail due to a default ESXi host certificate and its issuer not being in the trusted certificates lists. This is a combination of setting the issuer to ignore and not performing a revocation check. Setting this to false should be the first change to make when attempting to resolve heartbeat, RPC, or discovery issues to ESXi hosts when using PowerCLI versions later than 5.5.
- ESXi: Ignore Self-Signed Certs: If true, ignores any self-signed certs (subject = issuer) from ESXi hosts during heartbeat, RPC, and discovery.

Issues with self-signed certificates previously implemented by customers were caused by a security update to the VMware vSphere PowerCLI in versions after 5.5 that no longer permits the use of self-signed certificates.

Overview of Secret Launchers and Protocol Handlers

The relationship between Secret Server launchers and protocol handlers is integral to the functionality of Secret Server in automating and securing access to various applications and systems.

Secret Launchers

Secret launchers are tools within Secret Server that allow users to launch applications and automatically log in using credentials stored in Secret Server. There are several types of launchers, including:

- RDP (Remote Desktop Protocol)
- SSH (Secure Shell)
- Custom Launchers

These launchers provide a convenient method to open connections without requiring users to know or manually enter their passwords. For example, a web launcher can automatically log into websites using the client's browser.

Protocol Handlers

A protocol handler is an application installed on an end-user's machine that facilitates communication between Secret Server and the client machine. It also provides the necessary files for the launchers to function. When a user initiates a launcher, the protocol handler:

1. Bootstraps the client-side application.
2. Communicates with Secret Server over HTTP(S) to ensure it is the latest version and initiates an upgrade if necessary.
3. Bootstraps the target launcher type and begins the process of securely logging in the user.

The protocol handler ensures that credentials are retrieved securely from Secret Server using signed AES-256-encrypted messages, adding an extra layer of security.

Managing Multiple Instances

In environments with multiple instances of Secret Server, users might need to manage different versions of the protocol handler. To address this, the protocol handler auto-update function can be disabled, allowing users to manually update their protocol handlers as necessary.

Custom Launchers

Custom launchers extend the functionality of Secret Server by allowing integration with any application that can be started via the command line. They pass values from the secret text fields to the command-line of the application

being launched, enabling users to initiate processes or connect to services directly from the Secret Server interface without manually entering credentials.

Launchers

Launchers are tools within Secret Server that enable users to securely and conveniently access remote systems and applications using stored credentials without manually entering passwords. These launchers come in various types, including Remote Desktop Protocol (RDP) for Windows sessions, SSH for Unix systems, and web launchers for automatic website logins. The launchers work in conjunction with protocol handlers, which are applications installed on the user's machine to facilitate secure communication between Secret Server and the client machine. This setup ensures that credentials are retrieved securely using encrypted messages, adding an extra layer of security. Additionally, custom launchers can be created to integrate with any application that can be started via the command line, further extending the functionality of Secret Server.

Built-In Launcher Types

Secret Server launchers, supported by protocol handlers, come in three primary types:

- **Remote Desktop:** Launches a Windows Remote Desktop session and automatically authenticates the user to the machine.
- **PuTTY:** Opens a PuTTY session and authenticates the user to a Unix system.
- **Web Password Filler:** Uses a Chrome extension to automatically log the user into a website with secret credentials.
- **Web Launcher:** An alternative method to automatically log on websites. See "Web Launchers" on page 708.

Launcher Procedures

This section contains procedures for configuring Secret Server launchers.

Adding a Program Folder to the Windows PATH

If a launcher does not automatically add the program's folder to the Windows PATH:

1. Right click on **Computer** and go to **Properties**.
2. In the Properties window, click Advanced System Settings.
3. On the **Advanced** tab, click the **Environment Variables** button.
4. In the **System Variables** section scroll to **Path**.
5. Click **Edit** then at the very end of the text box, paste the full path to the folder where the program file is located, but make sure not to replace any existing entries. The list is semi-colon separated.
6. Click **OK** to close the dialogs.

Automatic Sudo or Su Privilege Elevation

Secret Server has a convenience feature that eliminates the need to manually enter a su or sudo command's password when using a proxied SSH session to a Unix or Linux server. When a user manually types a su or sudo

command with a valid secret ID, the SSH proxy automatically provides the username and password to use. The user does not need to know either.

For su, the connection procedure is as follows:

1. Secret A is created to contain the username and password for the su privilege elevation. Any potentially elevated users must have access to this secret.
2. Using secret B, a user (with access to secret A) starts a Secret Server proxied SSH session.
3. When the user types su at the command prompt, the SSH proxy detects it, determines the user has access to secret A, and augments the command with the secret ID for secret A via a command line argument. Any other arguments the user may have typed are left as is.
4. The user runs the su command, and the secret ID is replaced with the user and credential from secret A.
5. With the elevated permissions (temporarily as another user) from secret A, the user completes the desired tasks.
6. When finished, the user uses an exit command to return to their non-elevated status based on secret B.

> The added argument appears as `--secret-id <secret ID>` or `-id <secret ID>`, such as `su --secret-id 15`, which is replaced by a username and password from secret ID 15 when the command runs.

> Sudo does not take either secret argument and automatically types the current user's password.

Common Launcher Errors

Two of the most common launcher errors:

- **The process (process name) was not found:** The application has not been installed on the machine. If the application was installed, the program folder needs to be added to the path.
- **The stub received bad data (1783):** The process is set to launch as the credentials of the secret but the username or domain is not correct on the secret or the client machine cannot find the user or domain credentials specified.

Configuring Launchers on the Secret

Custom and SSH launchers provide additional settings on the Launcher tab of the secret for customizing authentication to the target.

- **Run Launcher using SSH Key:** If there is an SSH key set on the secret, it is used by default for authenticating to the target. Alternatively, you can specify a key from a different secret.
- **Connect As**: When an SSH secret is proxied, you can choose to connect as another user and then do an **su** to the current secret's user. This is a common practice for connecting with a lower privileged account and then switching to the root user.

Configure RDP Launcher Domain for Windows Account Template

Problem

When a Remote Desktop Launcher fails to log into a machine, it is sometimes because the machine is configured to have a default domain name (other than the local machine name). To determine whether the machine is configured to have such a default domain name, check one of the following:

Registry Setting: HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\DefaultDomainName

Group Policy (Windows 2008 and higher): "Assign a default domain for logon" under Computer Configuration\Administrative Templates\System\Logon

Solution

Add a key to `web-appSettings.config` that will cause the RDP launcher to use the machine name as the domain name when authenticating using a local Windows account.

1. Run a text editor as an administrator on the server running Secret Server.
2. Open the `web-appSettings.config` file located in the Secret Server application directory (typically `C:\inetpub\wwwroot\secretserver`).
3. Add the following key within `<AppSettings>`

   ```
   <add key="RDPUseComputerForDomain" value="true" />
   ```
4. Perform an IIS reset
5. Test the launcher.

Configuring SSH Proxies for Launchers

Launchers using an SSH connection can alternatively use Secret Server as a proxy rather than the launcher connecting directly to the target system from the machine it is being launched from.

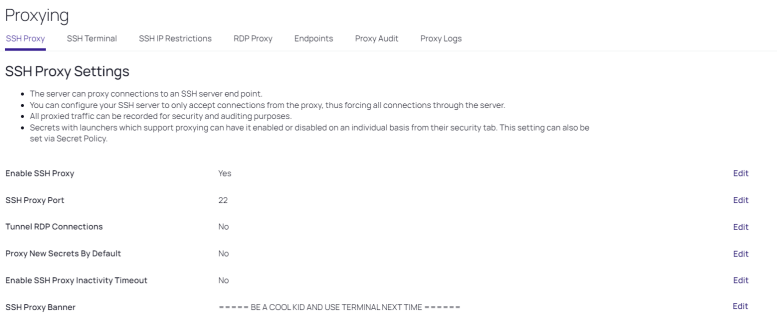
> Remote Desktop Services (RDS) is a special version of Secret Server Protocol Handler (SSPH) that can record keystrokes on its own, if configured in Secret Server. See "Secret Server Session Connector" on page 685 for details.

When proxying is enabled, all RDS sessions are routed through Secret Server. You can configure your SSH server to only accept connections from the proxy, thus forcing all connections through Secret Server. All proxied traffic can be recorded for security and auditing. You can enable or disable proxying for individual launchers. You can also do this using a secret policy.

In Secret Server Cloud, the distributed engine service also supports acting as a proxy for session launchers for greater network flexibility and offloading connections from the Secret Server instance.

To configure this:

1. Select **Admin > Proxying**. The SSH Proxy tab of the Proxying page appears:



2. Scroll down and click the desired **Edit** links to enter your SSH proxy configuration settings.

   The **SSH Proxy Settings** are:

   - **Enable SSH Proxy:** Enable or disable SSH proxying.
   - **SSH Proxy Port:** The port to proxy through. Changing this setting closes all active SSH proxy connections.
   - **Tunnel RDP Connections:** SSH Tunneling allows Remote Desktop Sessions to be proxied using the same proxy configuration settings. If enabled, RDP launchers will tunnel through a SSH Proxy if possible. This option predates the RDP Proxy which is now recommended instead.
   - **Proxy New Secrets by Default:** Enable proxying for applicable secrets when you create them.
   - **Enable SSH Proxy Inactivity Timeout:** Enable the SSH connection timeout - the period of inactivity after which the SSH will terminate the connection (15 minutes by default).
   - **SSH Proxy Banner:** Users connecting through SSH proxy see this text banner. This is not the same as the SSH Terminal Banner.
   - **Hide Passwords from SSH Keystroke Capture:** By default proxying records keystrokes. This disables that.
   - **Regular Expression to Find Password Prompts:** Specify regular Expression to find password prompts here. Note that modifying this field can slow down SSH Proxy output if regular expression is improperly formatted.
   - **Send Window Title Change Command on Startup:** Enable to change client window title to $USERNAME@$HOST.
   - **SSH Proxy Host Fingerprint**: The Secret Server SSH private key. Select **ECDSA** or **RSA** to generate the related SSH Proxy Host Key.
   - **Days to Keep Operational Logs**: Number of days to store operational audit logs.

   The **SSH Block List Settings:** SSH Proxy can block incoming clients that connect and fail to authenticate. Enable to select the following settings:

   - **Enable Block Listing:** Block incoming SSH proxy clients that connect and fail to authenticate.
   - **Auto Block Max Attempts:** How many times authentication can fail before the connection is blocked.
   - **Auto Block Max History:** How many times overall authentication can fail before the connection is blocked.

- **Auto Block Time Frame (minutes):** If you hit the maximum attempts within the time frame specified here, they are added to the block list. For example, if I have these settings set to five attempts and 30 minutes, if I fail to authenticate five times within 30 minutes I will be added to the list, but if I fail five times over five hours I would not be added to the list.

  Once the period has passed, the address must still be manually removed from the blocked list.

Client Override IP Address Ranges: IP address ranges that you can configure to always allow or always block the incoming connection. Click the **Add** link to add one.

- **Range:** Enter the IP address range, for example: 192.168.3.12, 192.168.42.147-192.168.42.194, 192.168.3.52/22.
- **Client Type:** Select Allow List or Block List.

Launcher Configuration and Support

Launcher configuration and support in Secret Server enable users to automate and secure access to various applications and systems by launching sessions directly from the Secret Server interface. Launchers can be configured for different types of connections, including Remote Desktop Protocol (RDP), SSH, and custom applications. The configuration process involves setting up the launcher type, mapping secret fields to the appropriate login fields, and enabling the launcher on the secret template. Secret Server supports both ClickOnce and MSI installer deployment methods for the protocol handler, ensuring compatibility with different environments. Additionally, advanced settings such as SSH proxy and "Connect As" commands can be configured to enhance security and flexibility. This robust launcher support streamlines the process of accessing remote systems while maintaining strict security controls.

Custom Launchers

In addition to the built in PuTTY and Remote Desktop launchers, Secret Server supports custom launchers. A Secret Server custom launcher is a feature that allows you to integrate Secret Server with any application that can be started via the command line. Custom launchers are designed to pass values from the secret text fields to the command-line of the application being launched. This enables users to initiate processes or connect to services directly from the Secret Server interface without having to manually enter credentials or other required information.

For more information on launcher arguments see "Custom Launcher Process Arguments" on page 674.

Like the built in launchers, custom launchers run on the users machine not on the web server. Launcher Processes can be set to run either using the credentials of the logged in user or the credentials of the secret. The "Run Process as Secret Credentials" check box is used to switch between theses two options.

Custom launchers are needed for several reasons:

- Integration with Various Applications: They allow Secret Server to work with a wide range of applications beyond the built-in launchers like PuTTY and Remote Desktop.

- Automation: They automate the process of logging into applications, saving time and reducing the risk of errors from manual entry.
- Security: They help maintain security by not exposing sensitive credentials, as the credentials are passed directly to the application without user interaction.
- Flexibility: They provide flexibility to tailor the launcher to specific organizational needs or to work with custom applications.
- Convenience: They offer a convenient way for users to access remote systems or applications with a single click from the Secret Server interface.

There are four types of custom launchers to choose from:

- **Process**: Launches a process on the user's machine and replaces parameters with values from the Secret.
- **Proxied SSH Process**: Launches an SSH client on the user's machine, connecting through Secret Server's proxy.
- **Batch File**: Launches a batch file on the user's machine using information from Secret Server.
- **Session Connector Launcher**: Allows for downloading and running an RDP file to launch into a Remote Desktop Server with a protocol handler installed.

Having four different custom launchers in Secret Server is necessary to provide flexibility and cater to a variety of use cases and applications that organizations might need to integrate with. Each type of custom launcher serves a specific purpose and addresses different requirements:

- **Process Launcher**: This is the most general type of custom launcher. It is used to launch any application that can be started from the command line on the user's machine. It is highly customizable and can pass parameters from the Secret directly to the application. This type of launcher is useful for a wide range of applications that do not require special handling or proxying.
- **Proxied SSH Process Launcher**: This launcher is specifically designed for SSH clients other than the built-in PuTTY launcher. It provides an additional security layer by proxying the connection through Secret Server, which prevents credentials from being exposed on the client's machine. This is particularly important for secure environments where SSH credentials need to be protected.
- **Batch File Launcher**: This launcher allows for the execution of batch files, which can contain a series of commands and can launch multiple processes. This is useful when a sequence of actions needs to be performed, or when integrating with complex systems that require more than a single command-line instruction.
- **Session Connector Launcher**: This launcher is used for more advanced scenarios where an RDP file needs to be downloaded and run to establish a Remote Desktop connection. It is particularly useful when client machines do not have certain applications installed, as it does not require any installation on the end-user's part.

The variety of custom launchers ensures that Secret Server can be adapted to the unique operational needs of different organizations. It allows for seamless integration with a diverse set of applications and services, enhancing both the user experience and security posture by automating access and protecting sensitive credentials.

See also "Configuring SSH Proxies for Launchers" on page 661.

Configuring WinSCP As a Custom Launcher

This topic is configuration only–see "General Settings" on page 669 for instructions.

Custom Process Launcher

General Settings

- Wrap custom parameters with quotation marks: No
- Record Multiple Windows: Yes
- Record Additional Processes: < None >
- Use SSH Tunneling with SSH Proxy: No

Windows Settings

- Process Name: winscp.exe
- Process Argument: `scp://$username:$password@$machine:$port`
- Process Argument: `sftp://$username:$password@$machine:$port`
- Run Process As Secret Credentials: No
- Load User Profile: No
- Use Operating System Shell: No

Process Arguments:

- WinSCP provides different protocols. The settings above show SCP or SFTP.
- `$machine` is used because it is a direct connection.

Custom Process Tunneled Launcher

General Settings

- Wrap custom parameters with quotation marks: No
- Record Multiple Windows: Yes
- Record Additional Processes: < None >
- Use SSH Tunneling with SSH Proxy: Yes

Windows Settings

- Process Name: winscp.exe
- Process Argument: `scp://$username:$password@$host:$port`
- Process Argument: `sftp://$username:$password@$host:$port`
- Run Process As Secret Credentials: No

- Load User Profile: No
- Use Operating System Shell: No

> Process Arguments:
>
> - WinSCP provides different protocols. The settings above show SCP or SFTP.
> - `$host` is used because it is a tunneled connection.

Custom Process Proxied Launcher

General Settings

- Wrap custom parameters with quotation marks: No
- Record Multiple Windows: Yes
- Record Additional Processes: < None >
- Use SSH Tunneling with SSH Proxy: Yes

Windows Settings

- Process Name: winscp.exe
- Process Argument: `scp://$username:$password@$host:$port /rawsettings Shell=/bin/bash`
- Process Argument: `sftp://$username:$password@$host:$port /rawsettings SftpServer=/usr/libexec/openssh/sftp-server`
- Run Process As Secret Credentials: No
- Use Operating System Shell: No

> Process Arguments:
>
> - WinSCP provides different protocols. The settings above show SCP or SFTP.
> - `$host` is used because it is a proxied connection.

SCP

`/rawsettings Shell=/bin/bash` must be specified to state the Unix shell used.

The value will be different depending on each account, `/bin/bash` is common but the exact value is obtained from executing the following command

```
[jeff@box ~]$ echo $SHELL
/bin/bash
```

SFTP

`/rawsettings SftpServer=/usr/libexec/openssh/sftp-server` must be specified to instruct how to launch the FTP server. The value could be different depending on what is installed on the Unix server–best to run one of the following commands to determine the answer:

```
[root@box jeff]# ps -ef --forest
```

```
root      7335  6171  1 11:26 ?        00:00:00  \_ sshd: jeff [priv]
jeff      7339  7335  0 11:26 ?        00:00:00      \_ sshd: jeff@notty
jeff      7340  7339  0 11:26 ?        00:00:00          \_ /usr/libexec/openssh/sftp-server
```

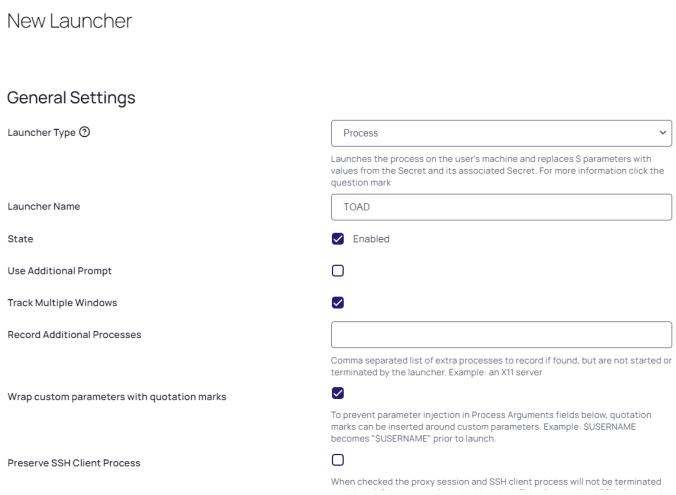
```
[root@box jeff]# ps -ef | grep ftp
```

```
root      3508     1  0 Jul18 ?        00:00:00 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
jeff      7340  7339  0 11:26 ?        00:00:00 /usr/libexec/openssh/sftp-server
root      7356  7250  0 11:29 pts/0    00:00:00 grep --color=auto ftp
```

Creating a Custom TOAD Launcher

You can create a custom launcher for TOAD by entering custom command line parameters in the "Process Arguments" field.

1. Navigate to **Admin > Secret Templates**,
2. Select the**Launchers** tab, and click **Create**.
3. On the New Launcher page, select a Launcher Type:
   - Use **Process** if you would like to use Secret credentials to connect directly to the remote host.
   - Use **Proxied SSH Process** if you have SSH Proxy enabled, to prevent passing Secret credentials to the client by connecting to Secret Server's proxy to interact with the remote host.



4. Enter a **Launcher Name** of your choice.
5. For **Process Name**, enter the location and the Toad executable. The location must exist on the client machine that will run the Toad launcher.
6. For **Process Arguments**, enter your own custom command line parameters, or the following:

   ```
   -C $USERNAME/$PASSWORD@$SERVER:$PORT/$DATABASE
   ```

7. When finished, click **Save**.

Creating and Implementing an Ultra VNC Custom Connection Launcher

Follow the steps below to create an Ultra Virtual Network Computing (VNC) custom connection launcher using Secret Server on a Windows machine.

Create an Ultra VNC Custom Connection Launcher

1. Navigate to **Settings > Secret Templates**. The Secret Template page appears.
2. Click the**Launchers** tab.
3. Click the **Create** button. The New Launcher page appears.
4. Click the **Launcher** type dropdown list and select **Process**.
5. Type a launcher name in the **Launcher Name** text box.
6. Go to the **Windows Settings** section.
7. Type `C:\Program Files (x86)\UltraVNC\vncviewer.exe` in the **Process Name** text box.
8. Type `$USERNAME $PASSWORD $HOST` in the **Process Arguments** text box.

   You may need to change the Process Arguments if the names of these fields in your Secret Template are something other than "Username" "Password" and "Machine"

9. Click the **Save** button at the bottom of the page.

Assign the Launcher to a Template

Assign the new launcher to an appropriate existing template. To build a new template specifically for VNC connections, see "Creating or Editing Secret Templates" on page 1171)

1. Navigate to **Settings > Secret Templates**.
2. Click the link for the desired template. The template's page appears.
3. Click the **Mapping** tab,
4. Click the **Add Mapping** button. The Add Mapping dialog box appears.
5. Click the **Mapping Type** dropdown list and select the launcher you created. New controls appear.
6. Click the **Domain** dropdown list to select the domain.
7. Type the password in the **Password** text box.
8. Type the username in the **Username** text box.
9. Click the **Save** button.

Creating Custom Launchers

This guide walks you through the process of creating a custom launcher, including defining launcher settings, and specifying command-line parameters. Follow the steps below to build a launcher that fits your organization’s unique needs.

Procedure

> See "Custom Launcher Errors" on page 671 if errors arise.

To create a new custom launcher:

1. Search for and select **Secret Templates**. The Secret Templates page appears.
2. Access the **Launchers** tab.
3. Click the **Create** button. The New Launcher page appears.
4. Configure the page as needed, see "Settings" below below for details.
5. Click the **Save** button.

> See the following section for details on General, Windows and Mac settings.

Settings

> Not all of the following settings are available for all types of launchers.

General Settings

The following settings are available in the General Settings section:

- **Launcher Type**: you can select either Process, Batch File, Proxied SSH Process, or the Session Connector Launcher.
- **Launcher Name**: Name of the launcher that is displayed to the user.
- **State**: Whether the launcher is active for use.
- **Use Additional Prompt**: If selected, the user is prompted for additional information when using the launcher. When selected, the **Additional Prompt Field Name** text box appears.

  > This option is mutually exclusive with **Use SSH Tunneling with SSH Proxy**.

- **Additional Prompt Field Name**: Name of the text field providing the prompt when the user activates the launcher. This value can be referenced in the process arguments with a `$` prefix.

After selecting a launcher type, a combination of the following additional fields appears:

- **Track Multiple Windows**: When this checkbox is selected, all visible windows of the primary process, not just the primary window of the primary process, are tracked. This helps record applications with multiple windows or dialog boxes. In addition, if the primary process (or one of its children) spawns child processes, any visible windows are recorded too.

  For example, if you run `cmd.exe` and then the `notepad.exe` application from the command prompt, notepad is recorded along with the command prompt. This checkbox is enabled by default. Enabling this setting is a prerequisite for **Record Additional Processes**.

- **Record Additional Processes**: Add a comma-separated list of additional process names to record if they are running. When a launcher is in progress and recording, any visible windows from the listed processes are also recorded. This only applies to processes running in your session—other users running the same process are not recorded. The processes themselves are not affected—they remain running after the launch is finished. This setting is only active if **Record Multiple Windows** is enabled too.
- **Wrap Custom Parameters with Quotation Marks**: This setting wraps the variables in the process argument fields with quotation marks. This is a security and disambiguation feature.

  For example, given these process arguments:

  ```
  --host=$HOST --port=$PORT --username=$USERNAME --password=$PASSWORD
  ```

  With no quotation mark wrap, the problematic process arguments for a launcher mapped to a secret might look like this:

  ```
  --host=xyz --port=123 --username=user --password=x x x
  ```

  The final parameter would be ambiguous, causing the last three characters to be misinterpreted, with the process thinking a single "x" is the password. Text could be injected, causing the value to be interpreted as another parameter, resulting in a security issue. Wrapping the parameter values fixes these potential problems:

  ```
  --host="xyz" --port="123" --username="user" --password="x x x"
  ```

  This checkbox is selected by default.
- **Preserve SSH Client Process**: When enabled, the proxy session and SSH client process will not be terminated and closed if the process launched is exited. This allows tabbed SSH clients to be used.
- **Use SSH Tunneling with SSH Proxy**: When enabled and the launcher is mapped to a secret template, the user can select host and port fields. You can provide a default port, which can be referenced using `$HOST` and `$PORT` in the process arguments. If the launching secret has proxying enabled and a Secret Server SSH proxy is available, those process arguments are replaced with SSH tunnel values (localhost [127.0.0.1] and a random port).

  This causes the launched process to connect to the local tunnel, and traffic then flows from the client to the Secret Server SSH proxy, which connects to the real endpoint. This is useful when users are not allowed to directly connect to the endpoint but Secret Server or distributed engines can.

  The checkbox is disabled by default. This option is mutually exclusive with **Use Additional Prompt**.

  > To use this feature, you must first map the host and port fields when you map the launcher to a template. To do so, edit the secret template and click **Configure Launcher**.

- **Use SFTP Tunneling with SSH Proxy**: Enable this setting when using SFTP to allow multiple data connections through the SSH Proxy. Many SFTP clients will not function correctly without this being enabled.

Windows Settings

The following settings are available in the Windows Settings section:

- **Process Name**: Name of the process that is launched. Example: `PowerShell`.
- **Process Arguments**: Process arguments depend on the process that is being launched. View the built-in SQL Server launcher for examples on how the text-entry fields are substituted. For greater flexibility, other secrets can be linked in the Launcher tab onto the secret. The text-entry field values from those secrets can also be used in the process arguments using the same prefix `$[1][FieldName]` syntax as the SSH custom commands.

  There is a launcher-specific token `$SESSIONKEY` that can be passed to the command line. This passes an identifier to the customer launcher that can be used to anonymously check-in the secret using the `CheckInSecretByKey` Web service method.

  Example: `-user $USERNAME -pwd $PASSWORD -f`. See "Custom Launcher Process Arguments" on page 674 for details.
- **Run Process as Secret Credentials**: This option allows the process to authenticate with the secret credentials (username, domain, and password) instead of the client user that is using the launcher. This can be overridden at the secret level to use a privileged account to run the process.
- **Use Operating System Shell**: this option allows for the use of the OS shell for the launcher. Useful for processes requiring UAC Confirmation.
- **Escape Character**: Enter the character to use as an escape character in passwords. Escape characters are required to allow the use of characters that are otherwise not allowed in passwords because they have special meaning to the launcher's target application.
- **Characters to Escape**: Enter the characters that require escaping for the target application.
- **Batch file**: Upload the batch file needed when the launcher is initiated.

Mac Settings

The following settings are available in the Mac Settings section:

- **Process Name**: Name of the process that is launched. Example: `/Applications/TextEdit.app/Contents/MacOS/TextEdit`.
- **Process Arguments**: Process arguments depend on the process that is being launched. View the built-in SQL Server launcher for examples on how the text-entry fields are substituted. For greater flexibility, other secrets can be linked in the Launcher tab onto the secret. The text-entry field values from those secrets can also be used in the process arguments using the same prefix `$[1][FieldName]` syntax as the SSH custom commands.

  There is a launcher-specific token `$SESSIONKEY` that can be passed to the command line. This passes an identifier to the customer launcher that can be used to anonymously check-in the secret using the `CheckInSecretByKey` Web service method.

  Example: `-user $USERNAME -pwd $PASSWORD -f`. See "Custom Launcher Process Arguments" on page 674 for details.
- **Shell Script**: Upload the script file needed when the launcher is initiated.

Custom Launcher Errors

Common errors when creating custom launchers:

The process (process name) was not found

The application has not been installed on the machine. If the application was installed, the program folder will need to be added to the path.

The stub received bad data (1783)

The process is set to launch as the credentials of the secret but the username or domain is not correct on the secret or the client machine cannot find the user or domain credentials specified.

Error(740): The requested operation requires elevation

When using "Run process as Secret credentials," even though the credentials have admin privileges, the process cannot be run with elevated privileges from the command prompt using runas. Instead, configure the process launcher as follows (substituting your .exe for program.exe):

- Process Name: cmd.exe
- Process Arguments: `/C start /B program.exe /wait`

Custom Launcher for SecureCRT (SSH)

The following instructions describe how to set up a custom launcher using SecureCRT:

Step 1: Creating the Custom Launcher

1. Log into Secret Server.
2. Search for **Secret Templates**.
3. Click the **Launchers** tab.
4. Click the **Create** button. The New Launcher page appears.
5. Click the **Launcher Type** dropdown list and select one of the following:
   - **Process**: use this type if you want to use secret credentials to connect directly to the remote host. This choice launches the process on the user's machine and replaces $ parameters with values from the secret and its associated secret.
   - **Batch File**: Not used for this task. Launches the indicated batch file on the user's machine. Allows the script to launch multiple processes using information from the server. Recommended only for advanced users.
   - **Proxied SSH Process**: If you have SSH proxy enabled, this type launches the specified SSH client on the user's machine. This prevents secret credentials from being passed to the client, by connecting to the Secret Server proxy to interact with the remote host. When the SSH proxy server is running, launched SSH sessions are proxied through the server.
   - **Session Connector Launcher**: Not used for this task. Allows for downloading and running an RDP file. This file is used to launch a Remote Desktop Server with protocol handler installed, making it unnecessary for end-user client machines to install anything. Recommended only for advanced users.

   > Depending on what launcher type you chose, all the steps below may not apply. The steps are in the order they appear in the UI, so if you do not see the item mentioned in the interface, you can ignore it.

6. For the sake of this example, choose the **Proxied SSH process** type.

7. In the **Launcher Name** text box, type the name *Secure CRT Proxied Process*.
8. Select the **State** checkbox to enable the launcher.
9. Select the **Use Additional Prompt** checkbox to add another field to the prompt. A text box appears to type the name of the field. You can reference the value in the arguments with the $ prefix.
10. Select the **Track Multiple Windows** checkbox to track child windows of the initial window.
11. Type a comma-delimited list of the names of other processes that are not started or terminated by the launcher, that you want tracked, in the **Record Additional Processes** text box. For example, an X11 server.
12. Select the **Wrap custom parameters with quotation marks** checkbox to prevent parameter injection in process argument fields. This means quotation marks are inserted around custom parameters prior to launch. For example: `$USERNAME` becomes `"$USERNAME"`.
13. Select the **Preserve SSH Client Process** checkbox to keep SSH client processes running after the launched process terminates. This is to support tabbed SSH clients and only applies to proxied SSH processes.
14. Select the **Use SFTP Tunneling with SSH Proxy** checkbox to enable using multiple SFTP data connections. Many SFTP clients require this setting to be enabled.
15. In the **Windows Settings** section, type the location and filename of the executable (`C:\program files\acme software\clients\securecrt.exe`) in the **Process Name** text box. The location must be on the client machine, i.e. the machine that will run the launcher.

    This step is a requirement for the SecureCRT launcher to work.

16. Type the following custom command-line parameters in the **Process Arguments** text box:

    ```
    /ssh2 /AUTH keyboard-interactive /PASSWORD $PASSWORD /P $PORT /L $USERNAME $HOST
    ```

    See "Custom Launcher Process Arguments" on the next page for details.

17. Click the **Save** button. The new launcher appears.

Step 2: Creating a Custom Secret Template (optional)

See "Creating or Editing Secret Templates" on page 1171 for details on creating a custom secret template.

Step 3: Associating the Launcher with a Secret Template

1. Log into Secret Server.
2. Search for **Secret Templates**. The Secret Templates page loads.
3. Click the link for the desired template. That template's page appears.
4. Click the **Mapping** tab.
5. Click the **Add Mapping** button. A popup appears.
6. In the **Mapping Type** dropdown list, select your custom launcher. The Host, Password, Port, and Username fields appear. For each field dropdown list, select the following:

- **Host**: <user input>
- **Password**: <blank>
- **Port**: <use default>. The **Default value** field appears.
- **Default value**: user's choice, pick an integer.
- **Username**: select an available item.

7. Click the **Save** button.

You can now launch SecureCRT whenever you use the launcher for secrets based off of this template.

Custom Launcher Process Arguments

Custom launcher process arguments can use a combination of parameters from:

- A field value from the secret.
- A field value from a linked secret.
- User input obtained from a prompt prior to launching.
- `$Host` and `$Port` (for use with a proxied SSH process or SSH tunneling)

For more information, see the "Dependency Token List" on page 1490.

Syntax

Parameters are prefixed with a dollar sign `$`. To obtain a value from the secret being launched, use `$FieldName`. To obtain a value from a prompt, use `$PromptName`. To obtain a value from a linked secret being launched, use `$[n]$FieldName` (where n represents the nth linked secret).

Examples

```
-user $UserName -color $[1]$Color -Location $LocationPrompt
-ssh $UserName@$Host -pw $Password -P $Port
```

Default Launcher Requirements

- **IBM iSeries Launcher:** For Secret Server itself, nothing is required. For distributed engines, the `ws3270.exe` file, found in the Secret Server directory, must be copied to the distributed engine folder on the engine, which is typically found at `C:\Program Files\Thycotic Software Ltd\Distributed Engine`. The `ws3270.exe` file itself can be downloaded on the ws3270 Downloads page.
- **PowerShell Launcher:** Requires PowerShell to be installed. When installed, the program is automatically added to the PATH.
- **SQL Server Launcher:** Requires SQL Server Management Studio to be installed. When installed, the program is automatically added to the PATH.
- **Sybase iSQL Launcher:** Requires that `isql.exe` is installed.
- **z/OS Launcher:** For Secret Server itself, nothing is required. For distributed engines, the `ws3270.exe` file, found in the Secret Server directory, must be copied to the distributed engine folder on the engine, which is typically

found at `C:\Program Files\Thycotic Software Ltd\Distributed Engine`. The `ws3270.exe` file itself can be downloaded on the ws3270 Downloads page.

Enabling CAC/PIV Smart Cards for Secret Launchers

Overview

A Common Access Card (CAC) or Personal Identity Verification (PIV) smart card is a physical card with an embedded electronic chip that uses a certificate-key pair to authenticate users. The certificate is issued by an authorized organization. The user has a PIN that should be known only to that user, which serves a second factor for two-factor authentication–access requires physical possession of the card, as well as the PIN. The user inserts the card into a card reader, which prompts for the PIN.

Secret Server launchers can pass smart card credentials through Remote Desktop Protocol (RDP) sessions. This is useful when a user needs to authenticate through an RDP session to a resource that requires smart card authentication, for example, a secured network drive that the user attempts to open while using the RDP session.

Currently, you can enable this either globally, via user settings, or per secret:

Enabling Globally with User Settings

1. In Secret Server, click the user icon and select **User Preferences**. The User Preferences page appears.
2. Click the **Settings** tab.
3. In the **Launcher Settings** section, click to enable the **Allow Access to Smart Cards** toggle. The change is automatically saved.

Enabling on a Specific Secret

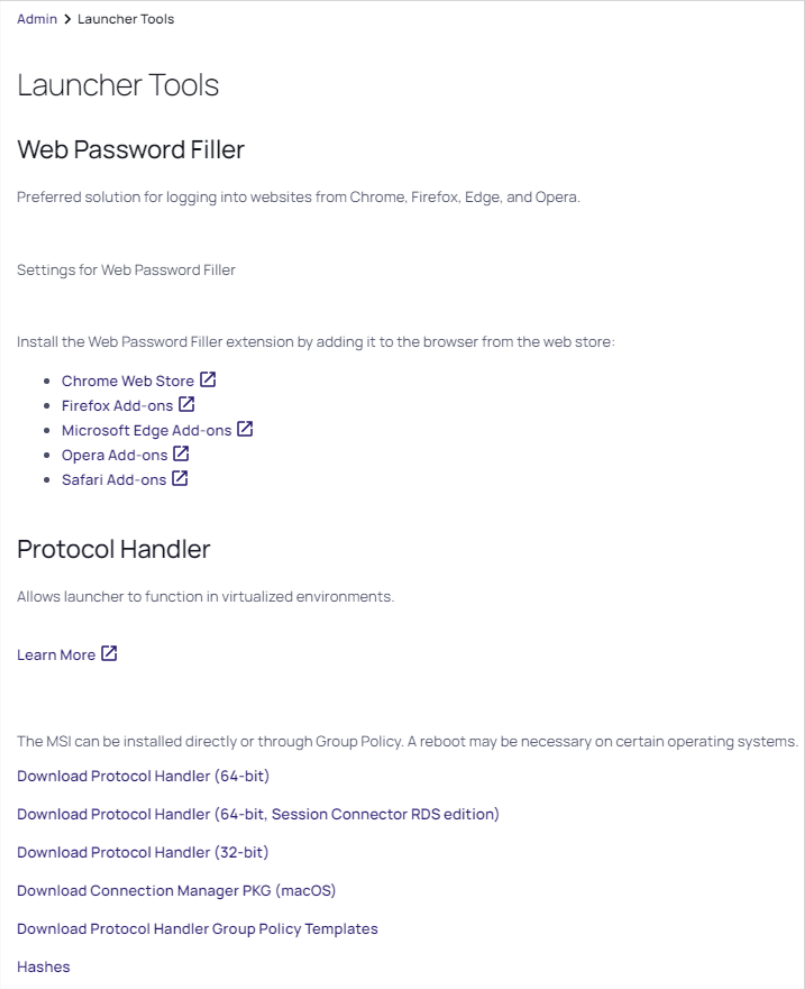
1. On a Secret with an RDP launcher, click the **Settings** tab.
2. Click the **Edit** link on the **Under RDP Launcher – Personalized User Settings** title bar. The page changes to edit mode.
3. Click to select the **Allow Access to Smart Cards** check box.
4. Click the **Save** button.

Enabling Launchers

Introduction

By default, the launcher is enabled by the **Enable Launcher** setting under **Admin > Configuration**.

The launcher (protocol handler) can be deployed in two ways–with the ClickOnce (the default) or MSI-installable applications. This can also be set in the configuration settings. The latter method allows the launcher to be used in virtualized environments or any environment in which the user does not have access to a Windows Temp directory. The Protocol Handler can be downloaded by clicking the 9-dot button on the Dashboard and selecting **Launcher Tools**:



A ClickOnce application is any Windows Presentation Foundation (.xbap), Windows Forms (.exe), console application (.exe), or Office solution (.dll) installed with ClickOnce technology in one of three ways: from a Web page, from a network file share, or from media. See ClickOnce Security and Deployment for details.

MSI Installer

To use the MSI installer (protocol handler installer) following steps below:

1. Select **Settings**.
2. In the **Tools and Integrations** section, click **Launcher Tools**.
3. In the section labeled **Protocol Handler Installer**, select the **Download Protocol Handler MSI** link for the operating system you want to install on.
4. Run the MSI file with admin privileges.

> The session is kept in check for security reasons with the session process pinging back to Secret Server to ensure it is still valid. This checks secret settings, such as checkout and secret access. If that check fails or the callback times out, Secret Server errs on the side of security and kills the sessions, ensuring access is not allowed.

Installing by Group Policy

The Protocol Handler application runs without requiring any input from the user. The installation may be pushed to your network without any special configuration. For details, see "Installing Protocol Handler Through Group Policy" on page 710.

Launching Sessions

On the Secret View page, clicking the Launcher icon launches the Remote Desktop, PuTTY, or custom session directly from the browser or log into the website. The mapped text fields are passed to the launcher for automatic authentication.

If the machine is set for Remote Desktop, the console launches and allows the machine to be specified from the RDP dialog.

If the Host is set to `<user>`, a prompt asks for the specific machine before launching the PuTTY session.

For some browser security levels, you might need to click **Allow** for the launcher application to open.

> The View Launcher Password permission can be removed to prevent users from viewing the credentials but can still use the authentication session to access the computer.

The settings under the Launcher tab are used for secrets that are enabled for SSH and custom launchers.

Limiting Launcher Domains

> This capability applies to Secret Server 10.9 or later. That is Secret Server Protocol Handler 6.0.0.28 or later.

You can limit the domains that a launcher connects to. If this is not set, then nothing changes–the launcher can connect to any domain. If it is set, however, Secret Server refuses to connect to any domains that are not explicitly allowed.

This setting is done via a Windows Group Policy Object (GPO) administrative template XML file (.admx). The file specifies the registry key that are changed when the GPO is edited. Download that file here: LimitLauncherDomainPolicyDefinitions.zip.

For details on using these files, see How to create and manage the Central Store for Group Policy Administrative Templates in Windows on the Microsoft site. The settings are present in both user and machine configurations in the group policy editor. If both are specified, then only the machine configuration is used (the user configuration is completely ignored). This is because the user configuration is stored in part of the registry that does not require administrator access to edit, so the machine configuration should be used in most cases.

The Group Policy valid values are just domain names, like `example.com`, or IP addresses, like `192.168.1.2`. No port should be specified, and no scheme. A value like `https://example.com` is not valid, because it has `https://`

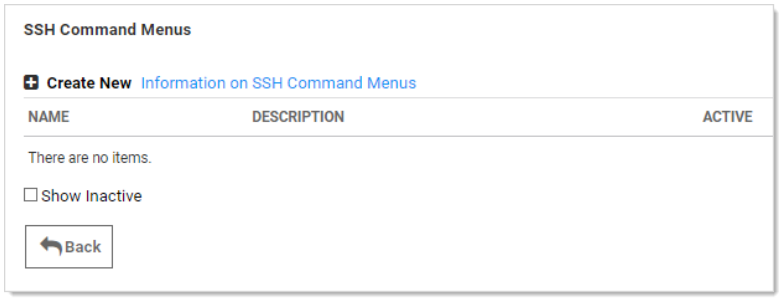
in the front. Ports are also invalid, so `example.com:885` will not match. The correct value would simply be `example.com`. Wildcards are not supported, but subdomains matter, so a value of `example.com` will not match `something.example.com`.

Managing Superuser Privilege

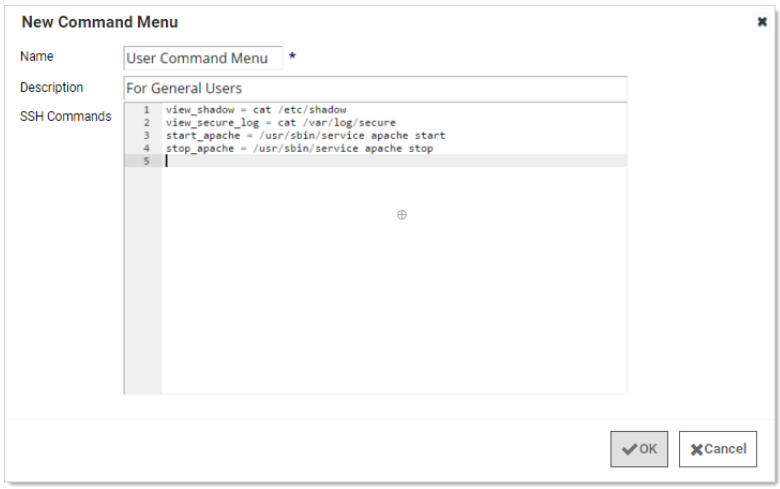
Administrators can create command menus for use with a proxied SSH connection to restrict what commands can be run by users or groups on the connected server. This feature requires an additional license. To add a command menu:

For details, see "SSH Command Menus" on page 856.

1. Navigate to **Admin > All**.
2. Click the **SSH Command Menus** button.



3. Click the **Create New** button.
4. Type a name, description and the SSH commands:

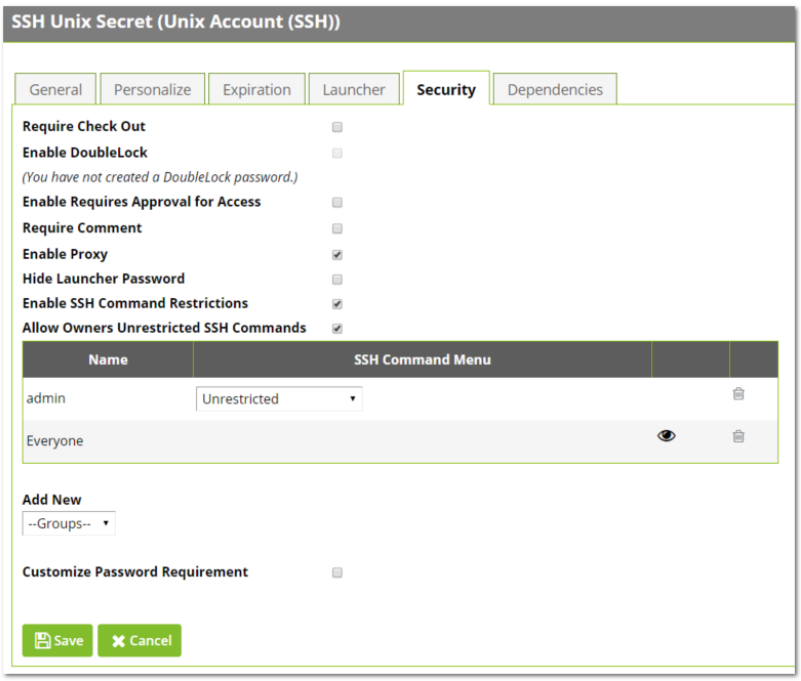


Once one or more command menus have been created, access can be controlled to individual Unix SSH secrets.

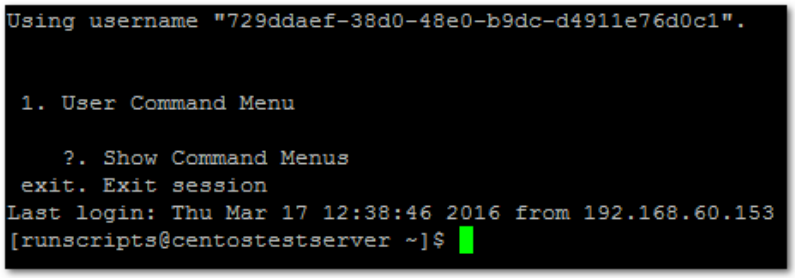
On the **Security** tab of a secret that can use a proxied PuTTY session, proxy must be enabled as well as command menu restrictions. If **Allow Owners Unrestricted SSH Commands** is enabled, any user who is an owner of the

secret has unrestricted use of the PuTTY session, that is, that user is able to type in commands as in a normal session. Additionally, other groups can be assigned the Unrestricted role as well.

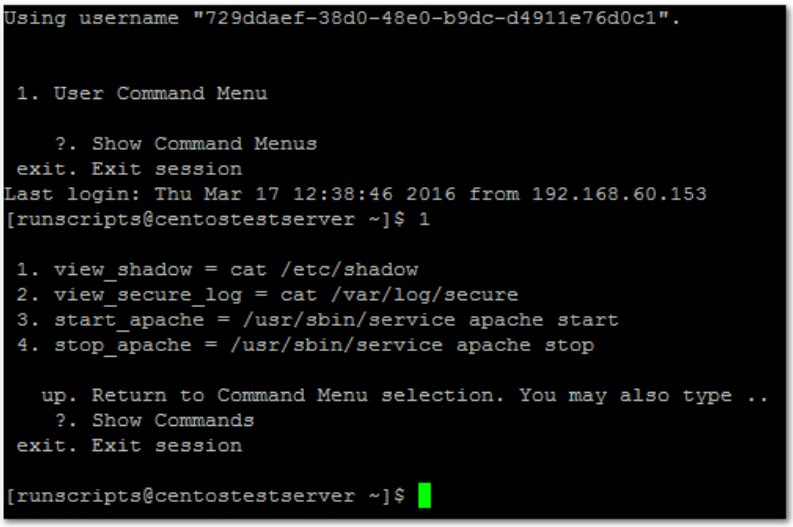
In the following example, the "admin" group is unrestricted, while everyone who is not in the admin group is restricted to only being able to run the commands that are enumerated in the user command menu, created above.



A user who is subject to SSH Command Restrictions are presented with a screen similar to the following when connecting to an SSH session:



The user simply enters the number of the command menu to see available commands, or types "?" to display the options again.



Only the commands listed can be run by this user. The user can either enter the number of the command to be run, or the name of the command, which is the word to the left of the equal (=) sign. Other options are available (as shown) to navigate through the available command menus, display help, or exit the session.

Remote Desktop Launchers

Secret Server remote desktop launchers provide a secure and convenient way to initiate Remote Desktop Protocol (RDP) sessions directly from the Secret Server interface using stored credentials. These launchers allow users to access remote Windows machines without needing to manually enter or even know the passwords, as the credentials are automatically retrieved and used from Secret Server. This not only streamlines the login process but also enhances security by minimizing the exposure of sensitive credentials. The launchers work in conjunction with protocol handlers, which facilitate secure communication between Secret Server and the client machine, ensuring that all credential exchanges are encrypted and secure.

Adding Remote Desktop Launchers

1. On the related template page select the Mapping tab, and click **Add Mapping** to add a launcher to the template.
2. On the following page, select mapping type and add Domain, Password, and Username. The text-entry fields below reflect the text-entry fields necessary to map to the launcher. In the case of a custom launcher, these text-entry fields are used to run the launcher process if the launcher is configured to run as secret credentials.
3. Choose a secret text-entry field in the drop-down menu on the right to map to each launcher value on the left. See the following section for further details on editing launcher configuration.
4. Click the **Save** button to add the launcher to the template.

Browser Configuration

Remote Desktop (RD) launchers require the following:

- **Firefox Configuration:** Firefox requires a helper add-on application to run the RD and PuTTY launchers. The Microsoft .Net Framework Assistant add-on and .NET framework version .NET 4.8+ needs to be installed.

- **Chrome Configuration:** If using ClickOnce, Chrome requires a Helper Add-on application to run the RDP and PuTTY Launcher. The ClickOnce add-on for Google Chrome Add-on needs to be installed. The launcher requires .NET framework version .NET 4.8+ as well.
- **SSL Certificates:** SSL must be set up properly for the RD launcher to work correctly. If Secret Server is using SSL certificates, they must be trusted at the user's computer. This is only an issue with self-created certificates.

Editing RD Launchers

On the related Secret Template page, under the Mapping tab, in the Launchers section, click **Edit** next to the Remote Desktop launcher to modify the settings for a launcher that has already been added to the template.

Click **Edit** next to the **Launcher mapping** section to define which fields from the Secret will be passed to the launcher: **Computer, Domain, Password, Username**. For a launcher to work properly, Secret Server requires credentials to be taken from secret text-entry fields. Fields must be assigned their corresponding credentials from the list. In addition to the secret fields, the domain can be mapped to `<blank>`, which passes an empty string to be used with local accounts, and the machine or host can be mapped to `<user input>`, which prompts the user for a specific machine to be used with domain accounts.

Click **Edit** next to the **Launcher restrictions** section to restrict values that can be passed to a launcher.

- **Restrict User Input**: Check to enable restriction. In cases where there are multiple endpoints to connect to, such as with a domain account, the machines can be restricted to a set list. You may specify a field of comma separated hosts or IP addresses that the user will be restricted to. The allow list will allow only these values, while the block list will allow all values except the hosts on the Secret Field. Example: 192.168.1.2, MACHINE.EXAMPLE.COM, 192.168.1.60
- **Use list fields**: Indicates where or not globally defined lists will limit the user input. Requires that this template has a field of type List or URL List.
- **Allow list**: User can select from values in this list.
- **Deny list**: User cannot enter these values. If used with an Allow List, the Deny List will take precedence.
- **Include machines from dependencies**: Check to add the list of machine names from the secret dependencies to either the allow list or block list.

RDP Launcher Settings

To enable or disable the specific RDP Launcher Settings, on a related Secret with configured RDP Launcher, under the Settings tab, click Edit in the RDP Launcher section. Here you can enable or disable the following settings:

- Connect to console - enable to allow connecting to console.
- Allow access to printers - enable to allow access to printers.
- Allow access to drives - enable to allow access to local drives.
- Allow access to clipboard - enable to allow copy to clipboard.
- Allow access to smart cards - enable to allow access to smart cards. See Smart Card Integration with Secret Server for more details.
- Use Custom Window Size - enable to set window height and width.

Setting Up Secret Templates for RD Launchers

Launchers can be accessed from any secret created from a properly configured template.

By default, the templates Windows Account, Active Directory Account, Cisco Account (SSH), HP iLO Account (SSH), Unix Account (SSH), Web Password, and SQL Server Account have the launcher configured.

Secrets can be configured for the launcher from within the Secret Template Designer page.

Clicking **Configure Launcher** displays the options available.

Removing the Mac Launcher

To upgrade or apply a fix to the Secret Server Mac Launcher, you must remove the version that is already installed. But first you must prevent the launcher from restarting, and terminate all processes related to the Delinea Launcher.

1. Open **Terminal** and type `launchctl remove com.thycotic.thycoticD`

   This step should remove the ThycoticDaemon and prevent the launcher from restarting, but you might need to perform the step more than once.
2. Open **Activity Monitor**.
3. Force Quit the **ThycoticLauncher** process and all related processes. See Quit an app or process in Activity Monitor on Mac.
4. Open **Finder**.
5. Navigate to **Application > Delinea**.
6. Right-click **ThycoticLauncher** and select **Move to Trash**.
7. Empty your trash.

Session Recording and Launchers

Session recording provides an additional level of security by recording a user's actions after a launcher is used. Session recording works for any launcher, including PuTTY and SSH, Windows Remote Desktop, Microsoft SQL Management Studio, and custom executables. The resulting movie is viewable from the secret audit. Session recording can be toggled on or off globally on the Configuration page and set for individual secrets on the Security tab. Detailed information on supported codecs can be found in "Session Recording Overview" on page 1226. When a user launches a session with session recording enabled, a brief message is displayed to inform the user that their actions are recorded.

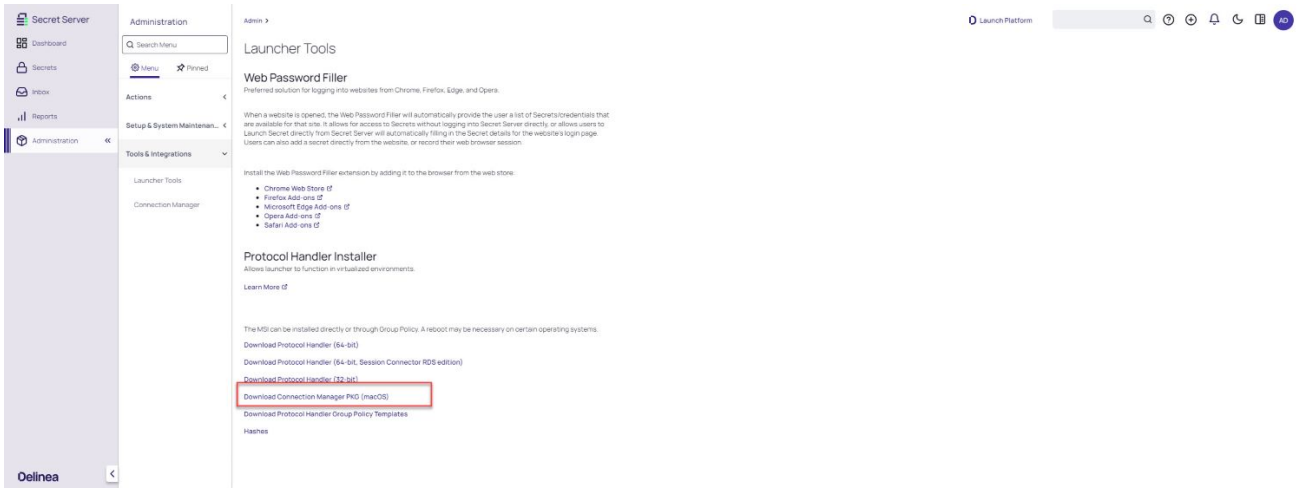
When multiple Launchers are enabled for a secret template, enabling session recording for a secret applies the setting to all launchers for that secret.

Protocol Handler cannot record processes when running as Secret Credentials without Administrator Elevation.

Setting Up the Mac Launcher

To set up the Mac launcher, simply follow these steps:

1. Inside Secret Server, navigate to Admin | Launcher Tools.
2. Select the Protocol Handler Installer for macOS



3. Navigate to your downloads directory and install the pkg on your Mac.

Removing the Mac Launcher

To upgrade or apply a fix to the Secret Server Mac Launcher, you must remove the version that is already installed. But first you must prevent the launcher from restarting, and terminate all processes related to the Delinea Launcher.

1. Open **Terminal** and type `launchctl remove com.thycotic.thycoticD`

   This step should remove the ThycoticDaemon and prevent the launcher from restarting, but you might need to perform the step more than once.
2. Open **Activity Monitor**.
3. Force Quit the **ThycoticLauncher** process and all related processes. See Quit an app or process in Activity Monitor on Mac.
4. Open **Finder**.
5. Navigate to **Application > Delinea**.
6. Right-click **ThycoticLauncher** and select **Move to Trash**.
7. Empty your trash.

Special Argument Handling

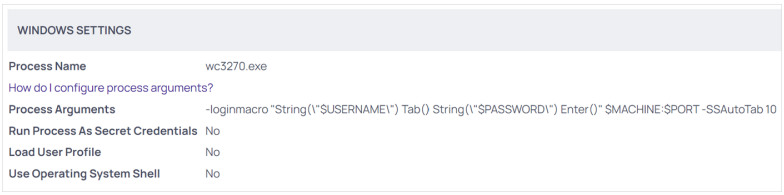
IBM iSeries Launcher

Usernames and passwords can run into length issues during the automated login process initiated by a launcher. The issue stems from the behavior in the client–when a user logs in and enters a username of 10 characters, the client automatically tabs to the next field (the password field). The launcher's automated login process automatically inserts a tab, which can cause improper behavior in the headless client.

If this behavior is observed, we added a parameter to the launcher's configuration to resolve it:

If you client adds a different number of tabs, replace 10 with that number.

1. Go to **Administration > Secret Templates > Launchers > IBM iSeries Launcher**.
2. Edit the value “Process Arguments,” adding the `-SSAutoTab 10` parameter: For example:



Using Connect As Command and SSH Proxy with a PuTTY Launcher

Overview

Connect As Command is an advanced setting for the PuTTY launcher type where SSH proxy automatically runs the su command for a Unix root account secret after the user launches a PuTTY session. This provides a user elevated privileges without allowing a remote root connection or giving the user direct access to the credentials.

The connection procedure is as follows:

1. An admin uses this instruction to set up secret A (a Unix root account secret) to use secret B (a regular Unix account secret) as its "connect as" secret.
2. A user launches secret A.
3. SSH proxy connects using secret B's credentials.
4. SSH proxy issues the su command to switch the user back to secret A.

The procedure is performed once at the beginning of the session.

As noted, to implement this feature, you typically use a Unix *root* account secret and a Unix *regular* account secret. The session usually launches as the Unix *regular* account secret that is specified in the **Secret To Use** field on a Unix *root* account secret's **Settings** page.

Setting up SSH Proxy to Use the Connect As Feature

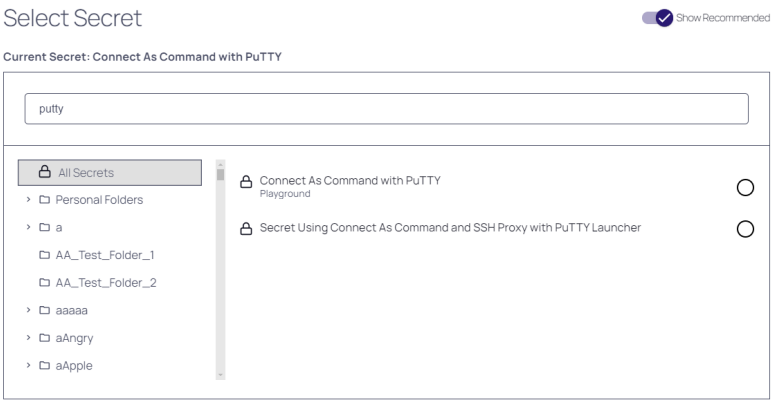
This procedure explains how to set a "connect as" secret when using SSH Proxy to allow connecting with a less privileged account and then using sudo or su to elevate privileges.

1. Make sure SSH proxy is enabled in Secret Server's global configuration settings.
2. Open a secret based on a template with SSH proxy enabled that specifies PuTTY as the launcher type to use.

   For this feature, we recommend building a custom secret based on a copy of the built-in **Unix Root Account (SSH)** template, and associating the PuTTY launcher with it.

3. Click the secret's **Settings** tab.
4. Click **Edit** in the **SSH Launcher** section. Next to **Connect Using**, select **SSH Key on another Secret**.

5. Next to **Secret to Use**, click **No Selected Secret**.



6. Select the related secret from the Select Secret list and click **Save**.
7. Navigate to **Admin > Secret Templates**.
8. Select the related template.
9. Under the Mapping tab, scroll down to the Launchers and click **Edit** next to the PuTTY launcher.
10. Verify that the commands in the **Connect As Command** field are correct.
11. Click **Edit** if you need to make changes.

Secret Server Session Connector

Overview

Introduction

Normally, Secret Server requires installing additional software such as Connection Manager or Secret Server Protocol Handler (SSPH) on the end-user computers to launch secrets, such as RDP, SSH, or custom, and optionally record the session.

With Secret Server Session Connector (SSSC) installed on a Remote Desktop Services (RDS) server, anyone who can download and launch a standard Remote Desktop Protocol (RDP) shortcut file can have the same experience. The RDS server itself runs a special SSPH for RDS–SSPH (RDS) as a remote app to record the sessions, so end-users do not need to install any additional software.

The SSSC feature is largely scalable and can be set up using a single RDS server, a load-balanced cluster of RDS servers, or multiple load-balanced clusters of RDS servers. Before you set up the SSSC feature, there are some baseline requirements for those RDS servers and on your domain.

SSPH (RDS) is sometimes referred to as RDPWin in this topic. RDPWin is the main executable that SSPH runs to launch and record sessions.

Table: Terms and Definitions

Term	Definition
RDP	*Remote Desktop Protocol*. A Microsoft protocol for remote control of computers.
RDPWin	The primary executable for SSPH.
RDS	*Remote Desktop Services*. Remote control services (using RDP) provided by a dedicated server or servers.
SSPH	*Secret Server Protocol Handler*. SSPH is an application on an end-user's machine. It enables communication between Secret Server and that client machine. It also provides the files needed by secret launchers.
SSPH (RDS)	*Secret Server Protocol Handler, RDS Version*. A special SSPH for use with SSSC that enables optional keystroke recording.
SSSC	*Secret Server Session Connector*. SSSC is the subject of this topic.

Session Connector Downloads

Most session-connector-related downloads are directly downloaded from Secret Server:

1. Click the **Settings** drawer on the main menu and select **All Settings**. The All Settings page appears.
2. Click the **Launcher Tools** link. The Launcher Tools page appears.
3. The download links are at the bottom of the page in the **Protocol Handler** Installer section.

The exception is the Secret Server session connector download.

The install files (.msi) are zipped to make them more download friendly.

Connection Sequences

Figure: Session Connector Connection Sequences for an RDS Server.

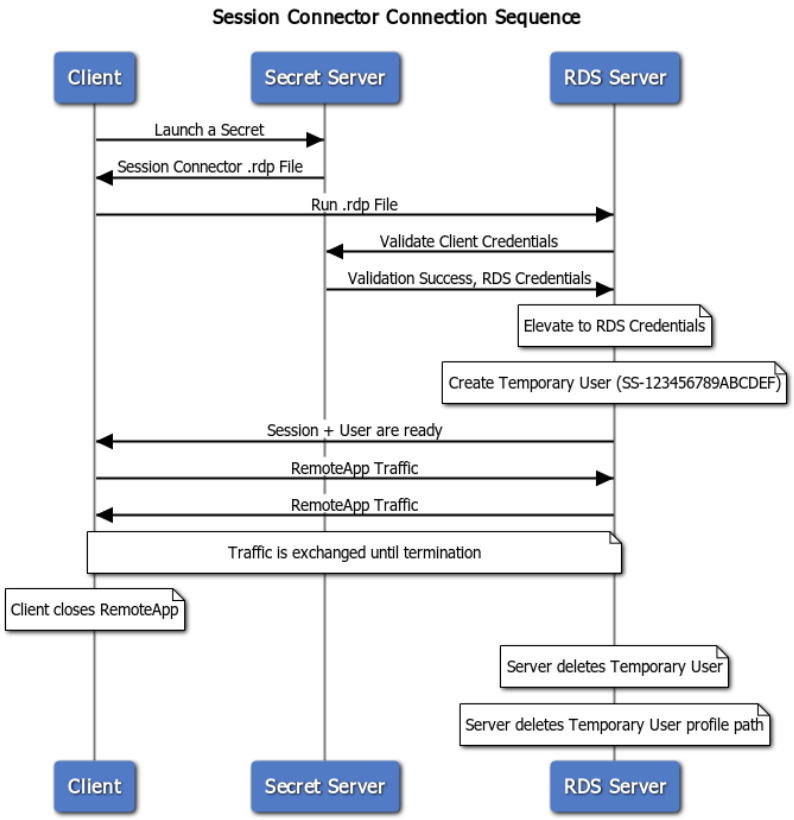
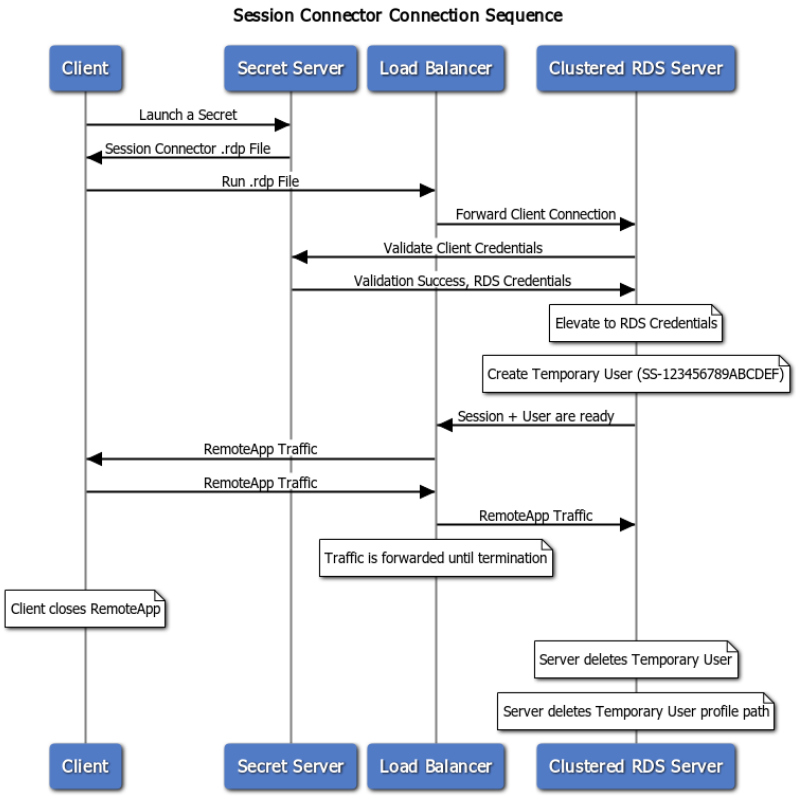


Figure: Session Connector Connection Sequences for Clustered RDS Servers.



Download

Session Connector is downloaded separately from Secret Server. Go to **Admin > Launcher Tools** to download the launcher application. Click the **'Download Protocol Handler (64-bit, Session Connector RDS edition) MSI'** link.

Configuration

> To comply with Microsoft licensing requirements, there is an additional constraint on which Microsoft Windows Server version you can use as the RDS server for session connector.

If you use Microsoft User Client Access Licenses (CALs), you cannot use Windows Server 2019. You must use Windows Server 2012 or 2016. If you use Microsoft Device CALs, you can use any supported version of Windows Server.

Task 1: Reviewing RDS Server Prerequisites

> Please ensure you turn on file and printer sharing on the Windows network configuration of the session connector machine (in the advanced sharing settings in the Network and Sharing Center) before RDS installation and configuration. This is because, when creating the collection, if such sharing is not on, the RDS service cannot recognize the session host by its name.

- Each RDS server should be a 64-bit installation of Windows Server 2012 or 2016.
- You **MUST** have access to the console session (non-RDP) to install the SSSC integration. This is in case of any of any errors during installation, which may disable RDP access to the server.
- Each RDS server must be domain joined. Configuration of the RDS feature requires being logged in as a domain user.
- Each RDS server needs to have a recent version of the C++ redistributable installed (v14.26.28720 or higher, May 2020):
  - Download: https://aka.ms/vs/16/release/vc_redist.x64.exe
  - More info: The latest supported Visual C++ downloads
- Each RDS server needs to have a credential available to manage temporary users. This credential should be able to create and delete local users and add users to the Remote Desktop Users group. If you plan to use one or more load-balanced clusters of RDS servers, this credential should be a domain user and will be used for all servers inside of a cluster. We recommend one domain user per cluster. This credential will be referred to as the **RDS Credential**
- Each RDS server needs to have the RDS Session Host Windows feature installed. See the next section.
- The Default Domain Policy should be set to Log on as a batch job. The SSSC dll requires this permission when it creates a disposable local account for the inbound session running the protocol handler. It calls the Win32 API LoginUserW passing in LOGON32_LOGON_BATCH as an input argument. Learn more about Logon user function here.

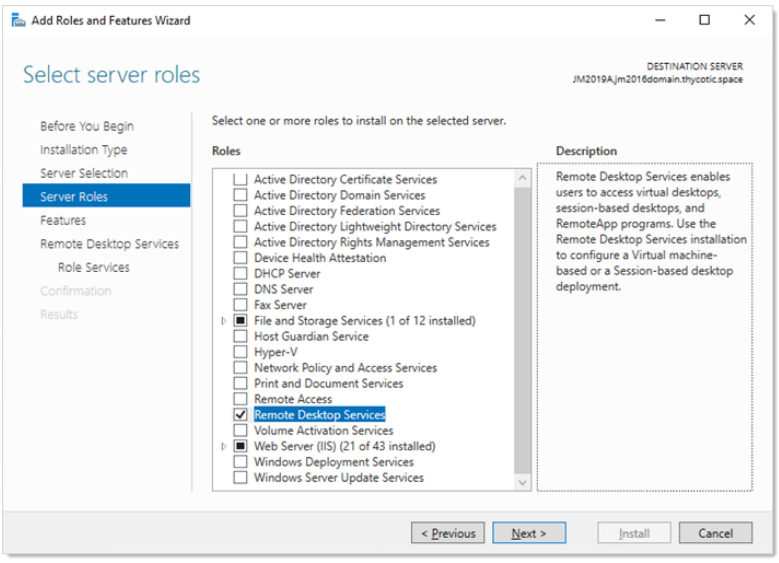
Task 2: Setting up RDS Services

Step 2.1: Installing Remote Desktop Services–Remote Desktop Session Host

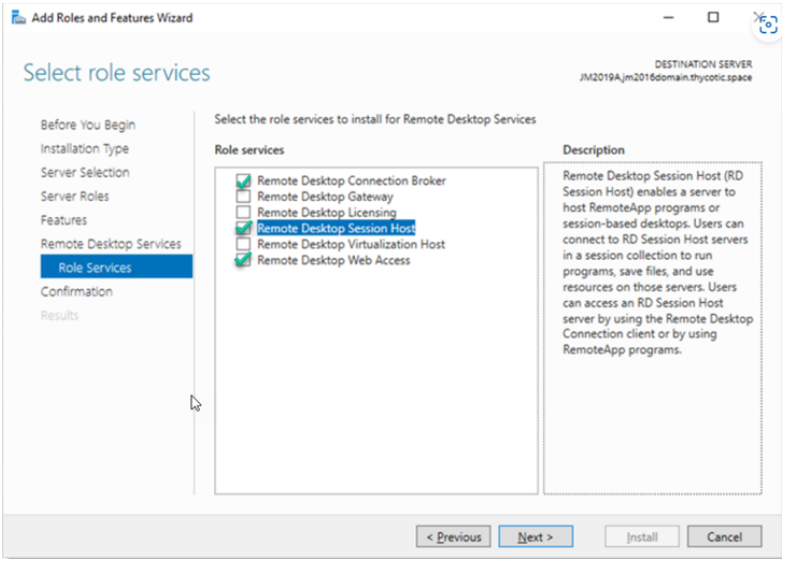
> SSSC cannot function without this feature and will refuse to install if it is not present. **RDS requires additional remote desktop licensing from Microsoft.** This may also require installing the remote desktop licensing feature if you do not already have a licensing server available in your environment. See Activate the Remote Desktop Services license server for details.

> The steps below have been tested on Windows Server 2012 or 2016. The steps in newer versions may vary.

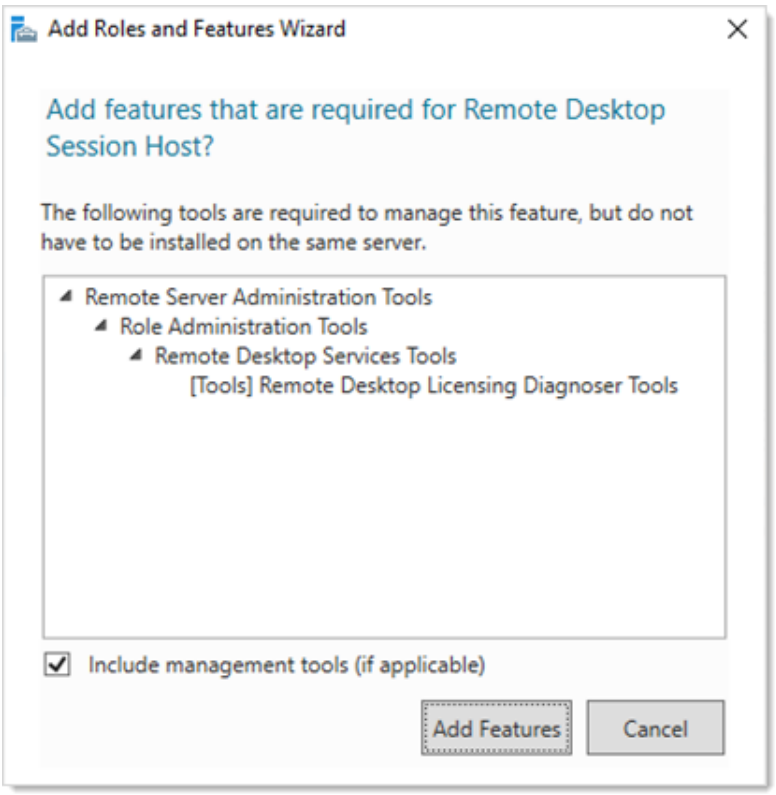
1. In Server Manager, click **Add roles and Features**. The Add Role and Features wizard appears.
2. Click the **Next >** button. The Installation Type page appears.
3. Select **Role-based or feature-based installation**.
4. Click the **Server Roles** menu item (or press **Next >** twice). The Select server roles page appears:



5. Click to select the check boxes for **Remote Desktop Services**, **Remote Desktop Connection Broker** and **Remote Desktop Web Access** roles.
6. Click the **Next >** button. The Select role services page appears:



7. Click to select the **Remote Desktop Session Host** check box.
8. Click the **Next >** button. The Add features... page appears:



9. Click the **Add Features** button. A "Confirm installation selections" page appears.
10. Click the **Install** button.

Step 2.2: Configuring Session Connector Settings

> The RDS session connector feature functions in the same manner without a certificate being set, but specifying a proper, password-protected certificate secret solves the trust issue between the publisher of the executed RDP file on the client and the RDS server. By doing this, you can avoid a warning indicating the “RemoteApp program can’t be identified” when beginning the session.

Enable Secret Server session connector:

1. Before continuing, note the following:
   - To create the certificate for this setting in secret server, create a new "Certificate with Private Key" secret.
   - The certificate file used by the secret must be of type .pfx or .p12 (PKCS#12).
   - The certificate file must be password-protected.
   - The certificate file password must be specified in the secret itself.
2. Go to `https://<your Secret Server or Platform location>/admin/configuration/advanced-config-settings`
3. Click the **Edit** button on the right side of the page.

> Do not change any other settings on this page without consulting Delinea Support. Your Secret Server installation could malfunction.

4. Click to enable the **Session connector enabled** check box.
5. (Optional) If you do not want to use the 900-second (15-minute) default, type your desired number of seconds in the **Session timeout** text box. Session connector .RDP files are valid for this many seconds (only for a single use). If set to 0 or below, the default is used..
6. (Optional) Click to enable the **Session Connector Allow Connection Sharing** check box. This changes the value of "disableconnectionsharing" in the output session connector RDP files. If true, this speeds up concurrent launches into the same RDS server quite a bit by re-using the existing Windows sessions at the risk of causing errors if launching a new session while an old session is in the middle of closing. The default is disabled.
7. (Optional) Select the **No secret selected** button to set a certificate secret. Setting this secret to a secret of type “Certificate with Private Key” enacts the process of signing the RDP file with a certificate for the Storage Area Network (SAN).
8. Click the **Save** button.

Step 2.3: Setting up RDS in Secret Server

1. Enable the **Session Connector** advanced configuration setting. For more instructions on this please follow the steps under **Configuring Session Connector Settings** below.
2. Go to **Admin > Configuration > General** tab.
3. Ensure the **Secret Server Custom URL** setting is set to a valid URL for your Secret Server. This URL is given out to SSPH launches (including SSSC and RDS SSPH) to ensure it knows how to connect back to Secret Server. Use HTTPS for maximum security. In fact, as of Secret Server version 10.9, SSPH and SSSC both refuse to connect to HTTP. This step applies to on-premise deployments only. Secret Server Cloud customers can skip this.
4. Create a Secret for the **RDS Credentials** mentioned above. If the credential is a local account, use a Windows Local Account secret, and if it is a domain user, use an Active Directory secret.
5. Create application users in Secret Server, one for each of the RDS server machines. See Creating RDS Application Accounts for details.
6. Share the secret created for the RDS credential mentioned above with the RDS application accounts that will be used by the RDS server(s). See Application Account RDS Credential Sharing.
7. Create SSSC custom launchers. For example, if you wanted to run an RDP session on the RDS server, you should configure a custom SSSC launcher that uses the built-in RDP launcher as its child launcher. See Configure Session Connector Custom Launchers.
8. Assign your SSSC custom launchers to the secret templates you want to launch from. See Assign Session Connector Custom Launchers to Secret Templates.
9. Configuration and setup is finished for Secret Server, but there are still some things you need to do inside of the RDS servers before setup is complete.

Task 3: Setting up RDS

Step 3.1: Installing the Secret Server RDS Protocol Handler

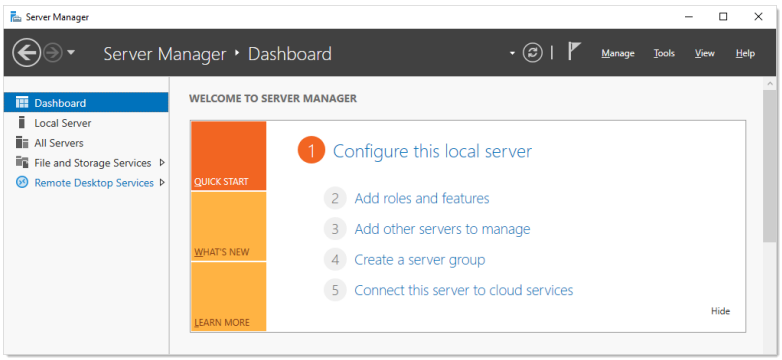
1. Go to **Tools > Launcher Tools** to download the launcher application. Click the **'Download Protocol Handler (64-bit, Session Connector RDS edition) MSI'** link.
2. (Optional) Ensure the listed hash value matches that for the file.

> SSPH (RDS) is a special version of SSPH that can record keystrokes on its own, if configured in Secret Server. Due to this optional keystroke recording, you may need to allowlist the `RDPWin.exe` file (the primary executable for SSPH) in any antivirus software running on the server. This is not currently necessary with Windows Defender.

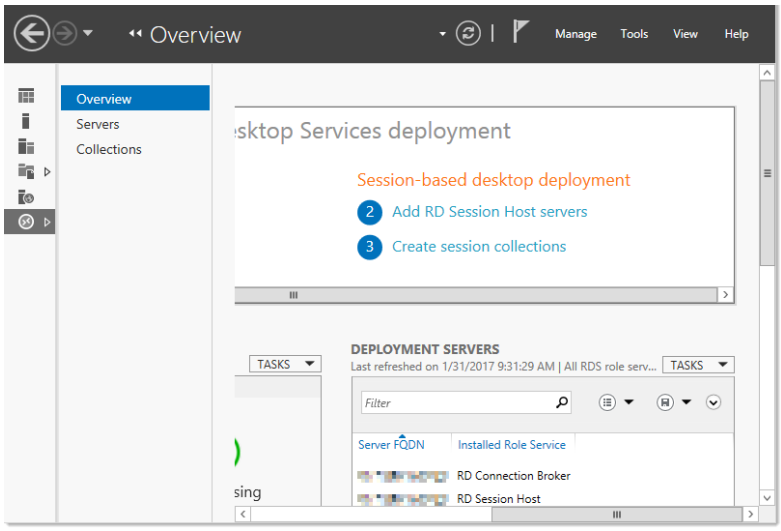
> SSPH (RDS) does not auto-update itself, unlike SSPH, because this could cause problems with multiple users running it at once on a single RDS server. Older SSPH (RDS) versions will continue to work with new Secret Server releases until updated, but a manual update is required on the RDS server(s) to take advantage of any future SSPH (RDS) features.

Step 3.2: Adding the Remote Desktop Collection and Application

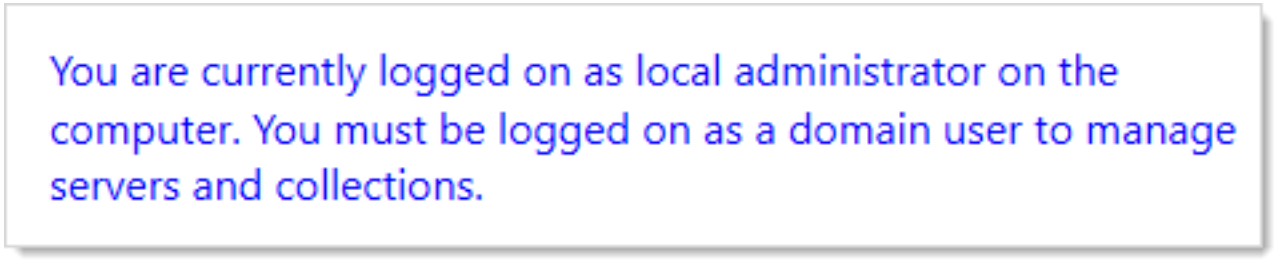
1. While logged in as a domain user, go to Server Manager:



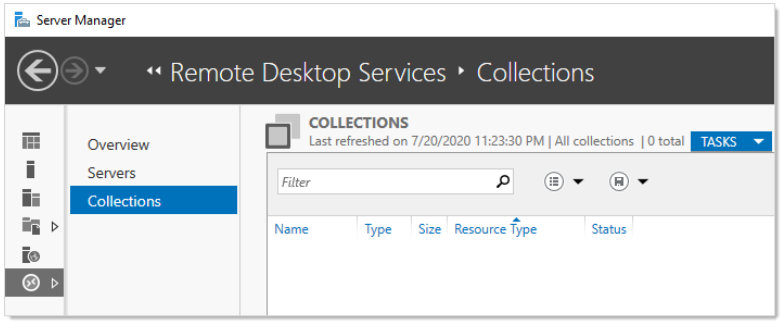
2. Click the **Remote Desktop Services** menu item on the left. The Overview page appears:



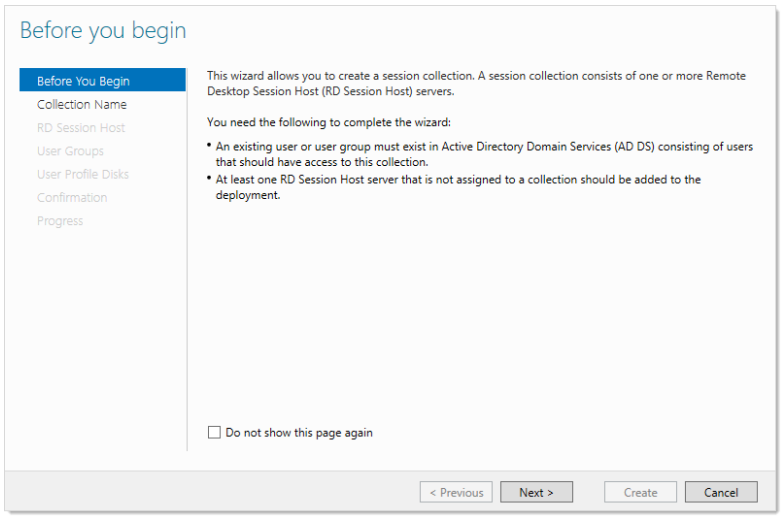
If you logged on as a local user, you will see this error and be unable to configure RDS. You must be logged on as a domain user.



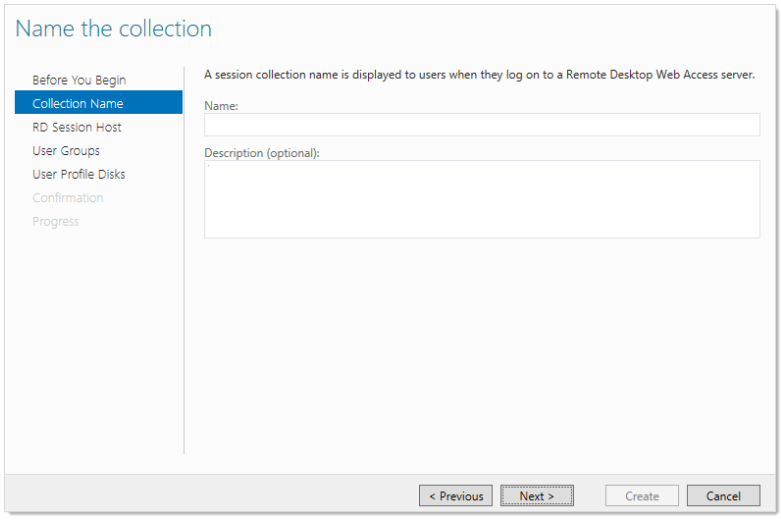
3. Click the Collections menu item. The Collections page appears:



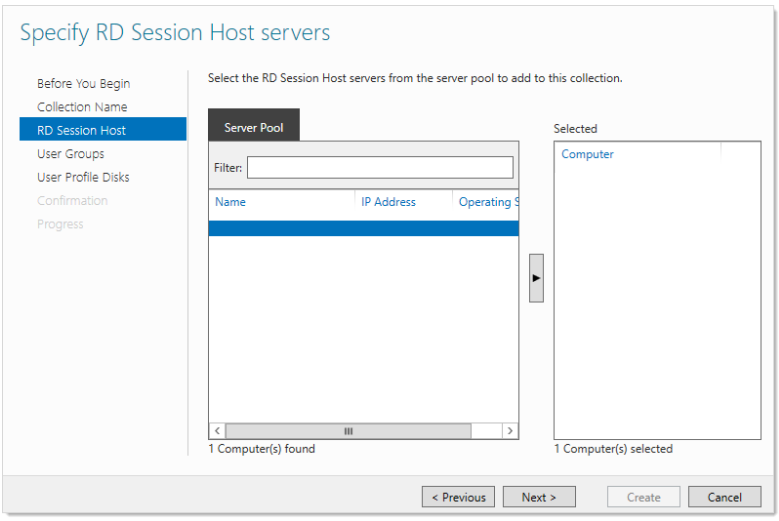
4. Click the **Tasks** dropdown list and select **Create Session Collection**. The **Create Collection** wizard appears on the Before You Begin page:



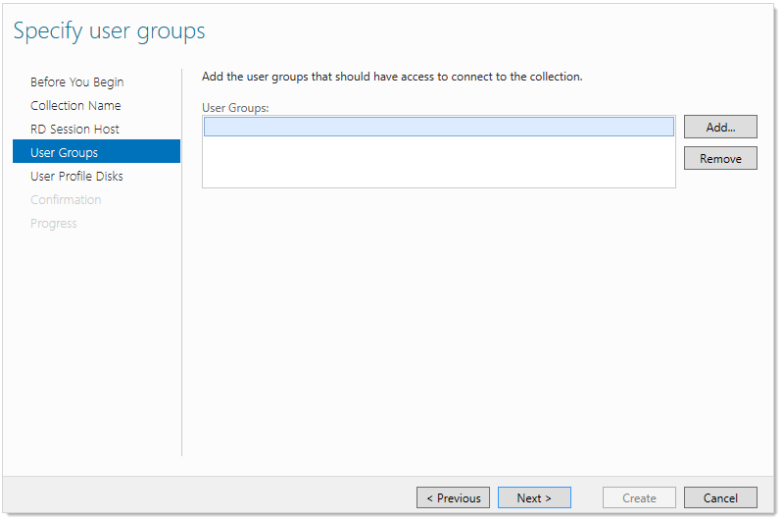
5. Click the **Next >** button to arrive at the Collection Name page:



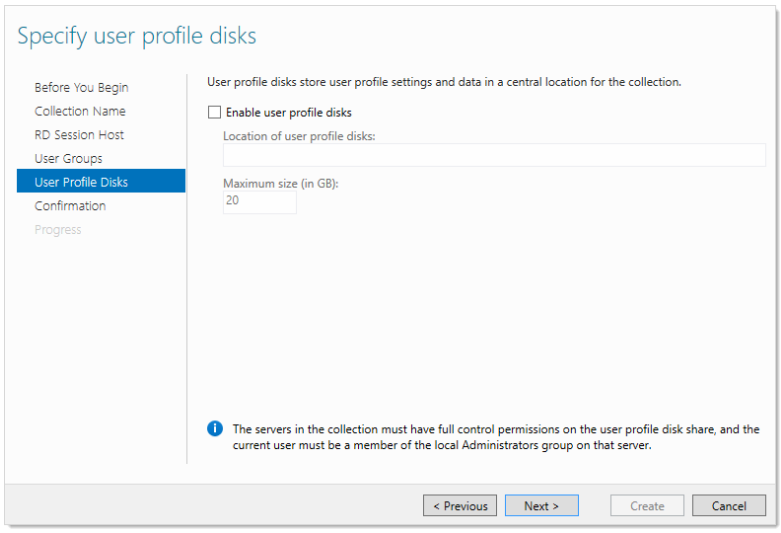
6. Type `Session Connector` in the **Name** text box.
7. Click the **Next >** button. The Specify RD Session Host Servers page appears:



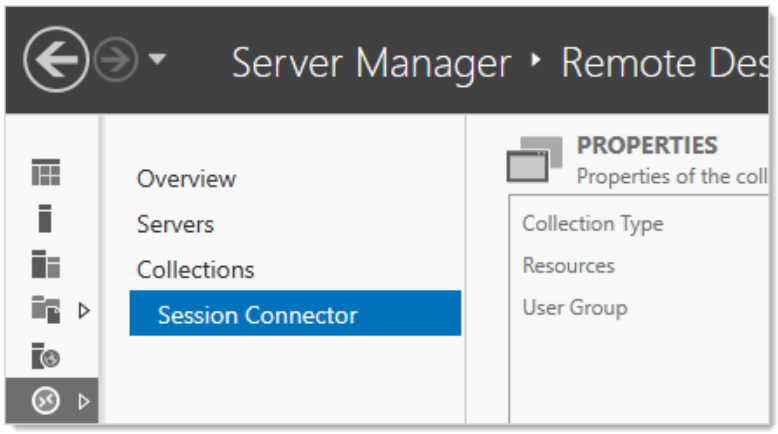
8. Add the local server from the left side (**Server Pool**) to the right side (**Selected**).
9. Click the **Next >** button. The User Groups page appears:



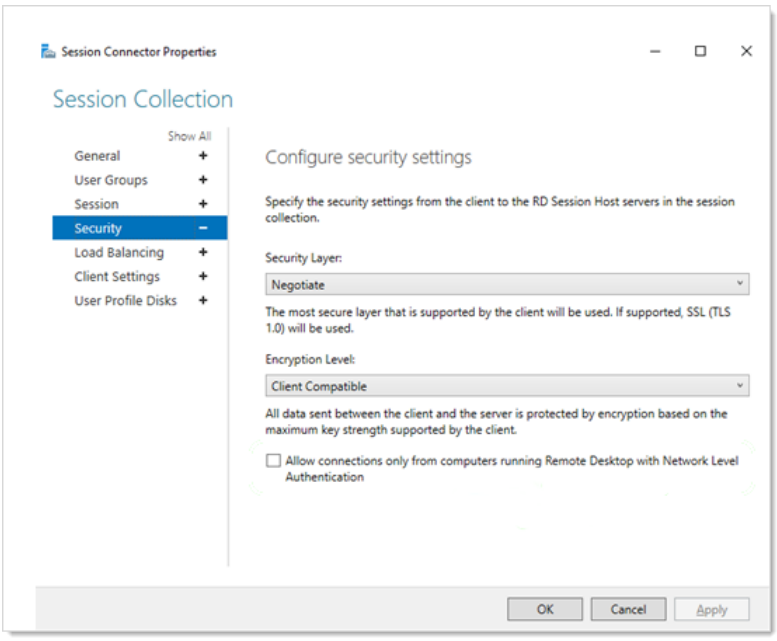
10. Select **Domain Users** in the **User Groups** list. This is not actually used by SSSC (it creates temporary local users), but RDS requires that something is selected.
11. Click the **Next >** button. The User Profiles page appears.



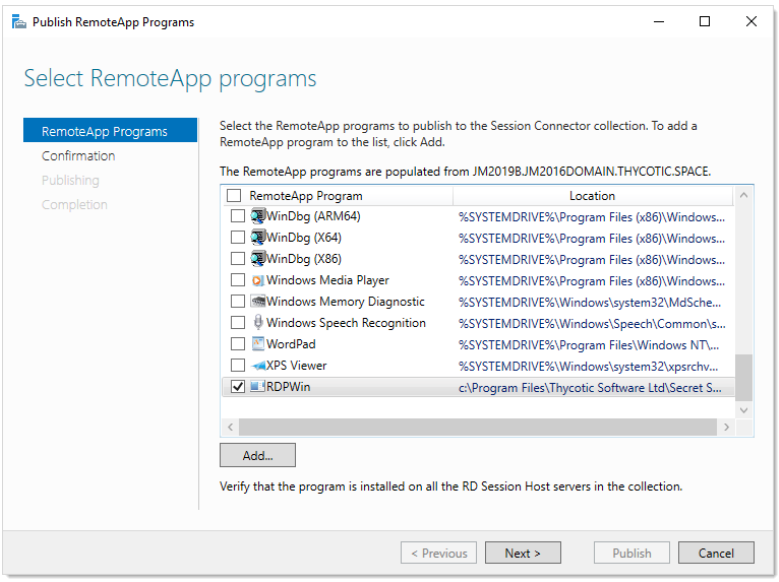
12. Click to select the **Enable user profile disks** check box to enable the **Create** button.
13. Click the **Create** button. The collection is created, and the wizard disappears. The session connector is now listed under Collections:



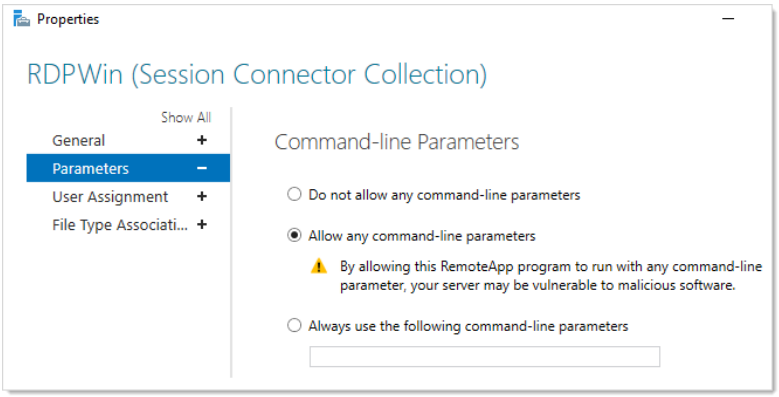
14. Click the **Tasks** dropdown list in the **Properties** section and select **Edit Properties**. The Properties popup appears.
15. In the left menu, click **Security**:



16. Click to deselect the **Allow connections only from computers…** check box. This necessary because SSSC uses temporary one-time use local users that do not exist until a connection is authenticated with Secret Server, making them incompatible with network-level authentication.
17. (Optional) If you want to restrict what can be mapped at the server level, such as drives, you can do so on the **Client Settings** tab. This is also configurable in Secret Server on each secret.
18. Click the **OK** button. The popup disappears.
19. Click the collection name in the menu on the left.
20. In the **RemoteApp Programs** section, click the **Tasks** dropdown and select **Publish RemoteApp Programs**.
21. Click the **Add…** button to add the RemoteApp for RDS protocol handler. A dialog box appears.
22. Navigate to `C:\Program Files\Delinea Software Ltd\Secret Server Protocol Handler`.
23. Select `RDPWin.exe`.
24. Click the **Open** button. The dialog closes, and RDPWin (SSSH (RDP)) is now selected in the list:



25. Click the **Next >** button. The Confirmation page appears.
26. Click the **Publish** button to save.
27. Click the **Close** button.
28. On the RemoteApp Programs page, right click **RDPWin RemoteApp** and select **Edit Properties**. A property panel appears.
29. Click the Parameters menu item on the left:



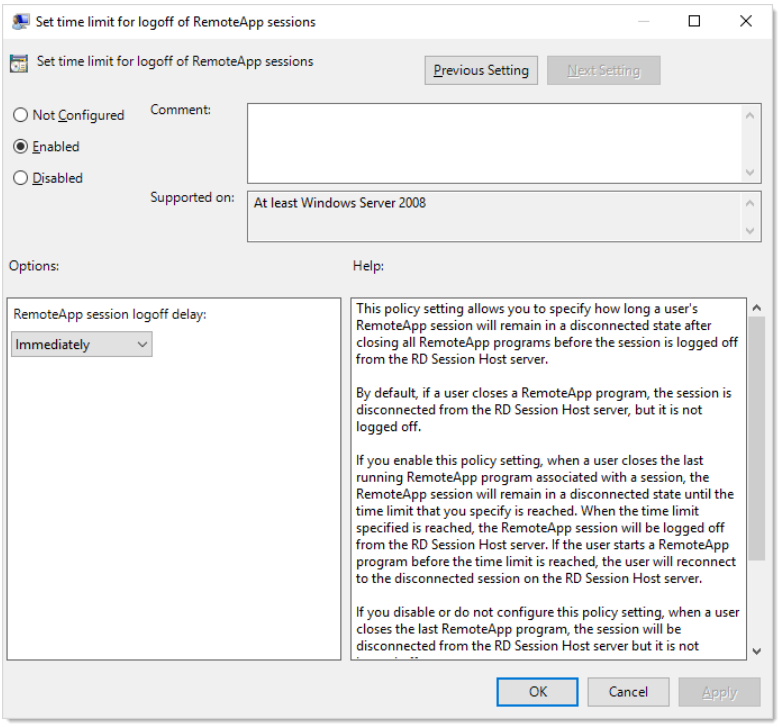
30. Click to select the **Allow any command-line parameters** selection button.
31. Click the **OK** button.

Step 3.3: Configuring RDS-related Group Policy Settings

To configure on a single server:

Set Time Limit for *Logoff of RemoteApp Sessions*:

1. Run the Group Policy Editor (gpedit.msc).
2. Go to **Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Session Time Limits**.
3. Click **Set time limit for logoff of RemoteApp sessions** to edit it. Its properties appear:



4. Click the selection buttons to select **Enabled**.
5. Click the **RemoteApp session logoff delay** dropdown list and select **Immediately**.

Disable the *Always Prompt for Password Upon Connection* GPO Setting:

1. Navigate to **Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Security**.
2. Locate the policy **Always prompt for password upon connection**.
3. Set this policy to Disabled. This ensures that passwords are automatically filled when launching an .rdp file, which is necessary for the seamless operation of the Session Connector.
4. Click **OK** to save.

Task 4: Updating API Credentials

The credentials for the SA application account are saved encrypted in the registry. The credentials are restricted to the NETWORK SERVICE account, which Remote Desktop runs under using DPAPI-NG.

If those application account credentials change in the future, follow these steps to update them:

1. Run the Windows Registry Editor, `Regedit.exe`.
2. Navigate to `HKLM\SOFTWARE\Delinea\SessionConnector`.
3. Set **CredentialsEncrypted** to 0.
4. Set **SecretServerUsername** to the plain text new username.
5. Set **SecretServerPassword** to the plain text new password.

These credentials are encrypted upon their first use, either the next time someone launches a SSSC session that hits this server, or if you reboot the entire server. Once this happens, returning to the Registry Editor, CredentialsEncrypted will be set back to "1," and an encrypted version of the username and password will be visible.

Task 5: Launching Session Connector Sessions

Now that it has been configured and installed, you should be able to launch SSSC sessions.

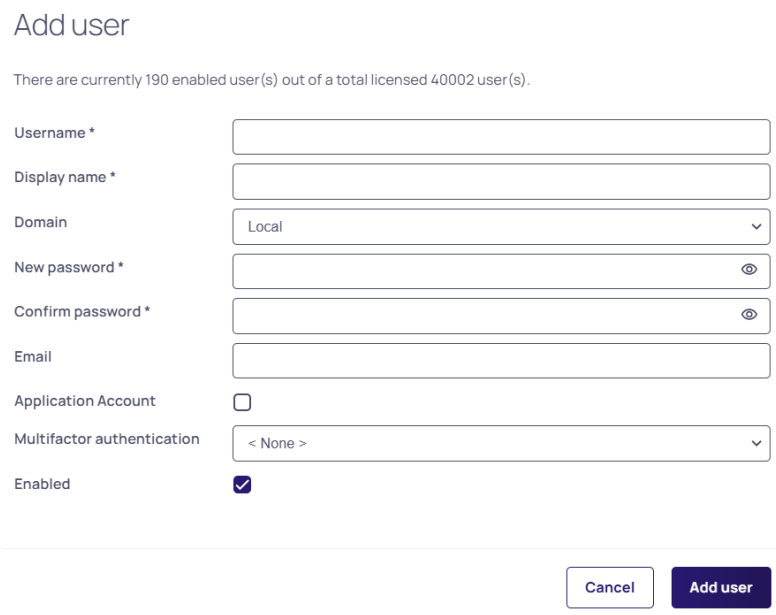
Once configured, the SSSC custom launchers appear just like any other launcher on the associated secret template types. When clicked, a Remote Desktop shortcut (.RDP) file is downloaded. This .RDP file can then be opened by standard Remote Desktop clients, such as mstsc.exe in Windows or RoyalTS in OSX.

When launched, the end-user will connect to the RDS host configured on the SSSC custom launcher. The RDS host then launches the RDS protocol handler and connects to the actual destination machine.

Subprocedures

Creating RDS Application Accounts

1. Go to **Admin > Users**.
2. Click the **Create User** button. The Add User page appears:



3. Type in or set the account details.
4. Ensure that the **Enabled** check box is selected.

> Leave the Multifactor authentication set to default <None, because this account is for SSSC and not a human being, so Multifactor authentication is not appropriate.

5. Click to select the **Application Account** check box. As an application account, the user can only log on through the application account API and does not require a separate user license.

> We recommend application account users because only API access is required by SSSC, and they do not consume regular user licenses. You may want to name the users to make it obvious which server they belong to. We recommend one user per RDS server for auditing purposes and to avoid one server with invalid credentials locking out the user, impacting all the other servers. See "REST API Reference Download" on page 1500 for more about the API.

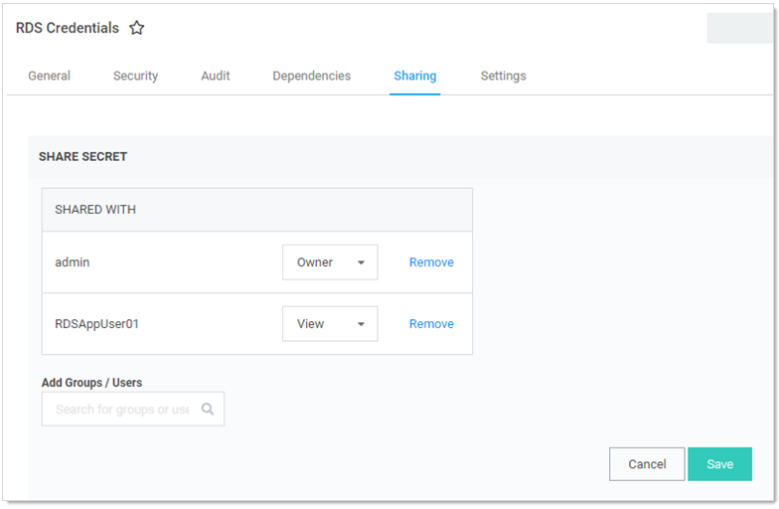
6. Click the **Save** button.
7. Repeat this process for each RDS server if you are clustering more than one.

> Application account needs to be added to "Logon as Batch job" GPO on the RDS servers.

Enabling Application Account RDS Credential Sharing

Each RDS application account must have view access to the RDS Credential that the RDS server(s) use to manage the temporary Windows local accounts:

1. Go to the RDS credential secret you created earlier.
2. Click the **Sharing** tab:



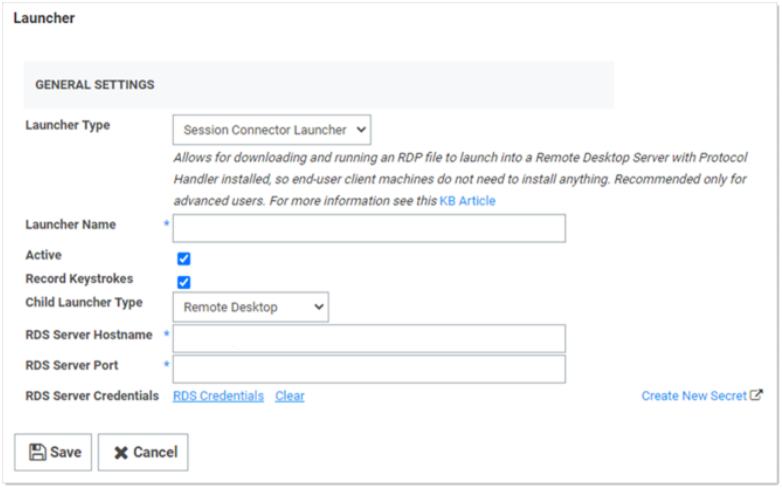
3. Grant view access to the applicable application account users. That is usually one Secret Server user account per RDS server. If you are using a cluster, this secret would be an Active Directory secret for a domain

credential that all the RDS servers can use, and you would share it with each of the RDS application accounts for each RDS Server in the cluster using in their SSSC configuration.

Configuring Session Connector Custom Launchers

You must create a custom launcher for each combination of and RDS server cluster and custom launcher type:

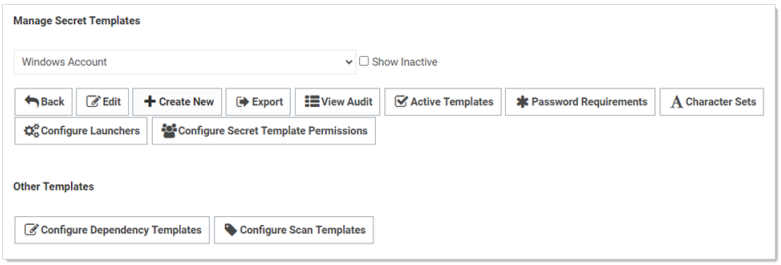
1. Go to **Admin > Secret Templates**.
2. Click the **Configure Launchers** button. The Launcher Types page appears.
3. Click the **New** button. The Launcher page appears:



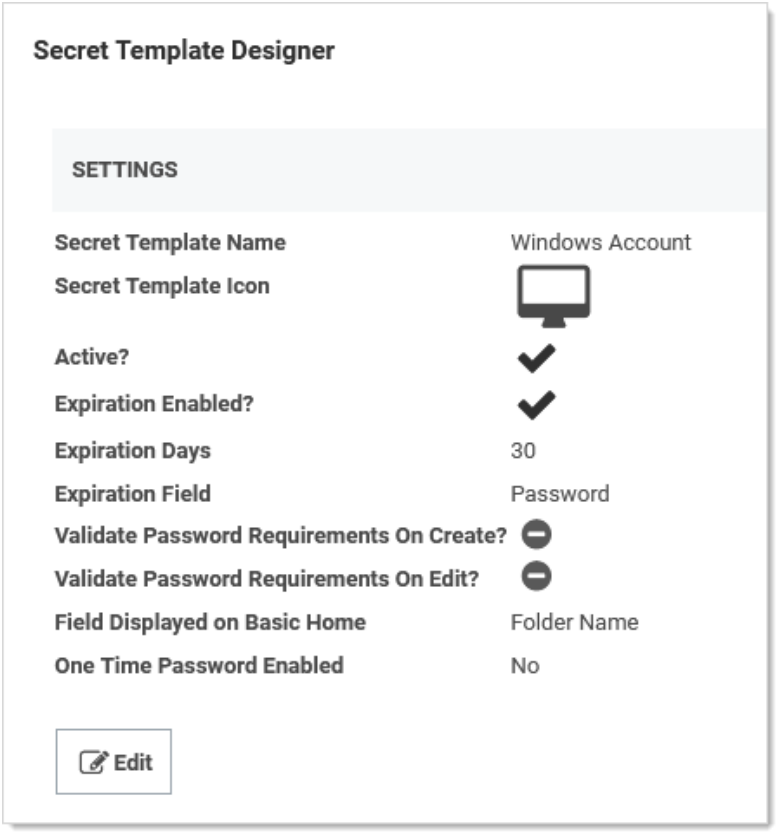
4. Type or set the parameters as follows:
   - **Launcher Type:** Session Connector Launcher. This launcher type will not be visible until the Configuration Advanced Setting is enabled.
   - **Active:** Ensure this is selected.
   - **Record Keystrokes:** Check to record keystrokes in addition to video on related secrets with session monitoring enabled.
   - **Child Launcher Type:** Click to select the launcher type, such as Remote Desktop or PuTTY. This is the real launcher type that runs on the RDS server to connect to the secret.
   - **RDS Server Hostname:** IP or hostname for the RDS server or cluster.
   - **RDS Server Port:** Type the port. The default RDP port is TCP 3389.
   - **RDS Server Credentials:** Click the **RDS Credentials** link to pick the Secret configured above for credentials that can create and delete local users. If RDS Server Hostname points to a cluster, all servers must be able to use these credentials.
5. Click the **Save** button.

Assigning Session Connector Custom Launchers to Secret Templates

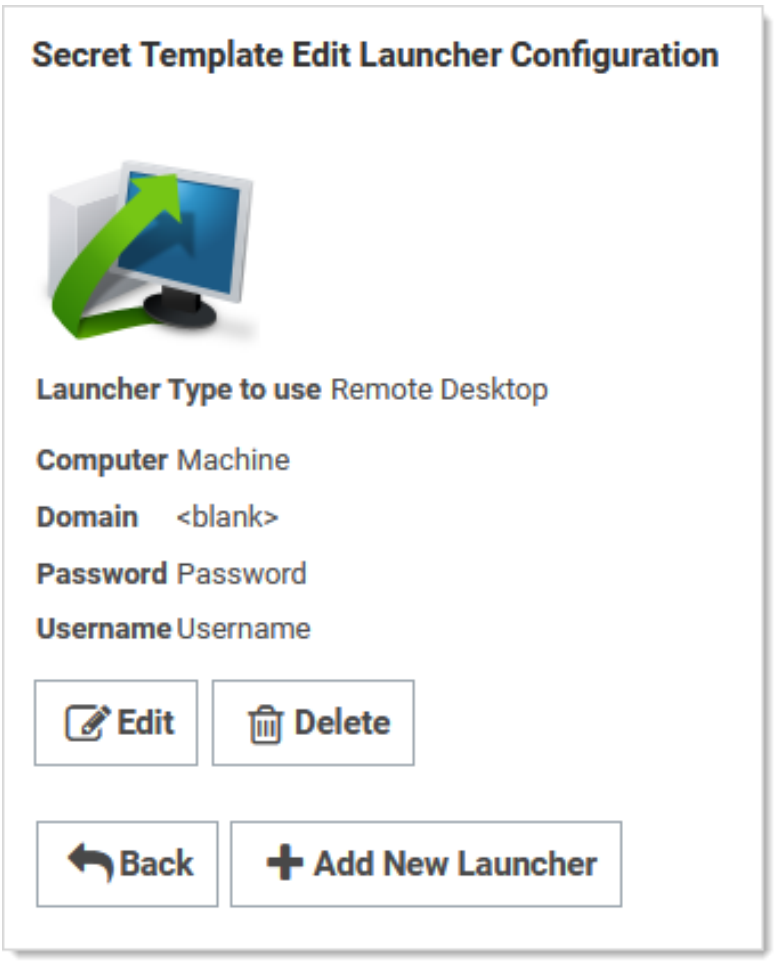
1. Go to **Admin > Secret Templates**. The Manage Secret Templates page appears:



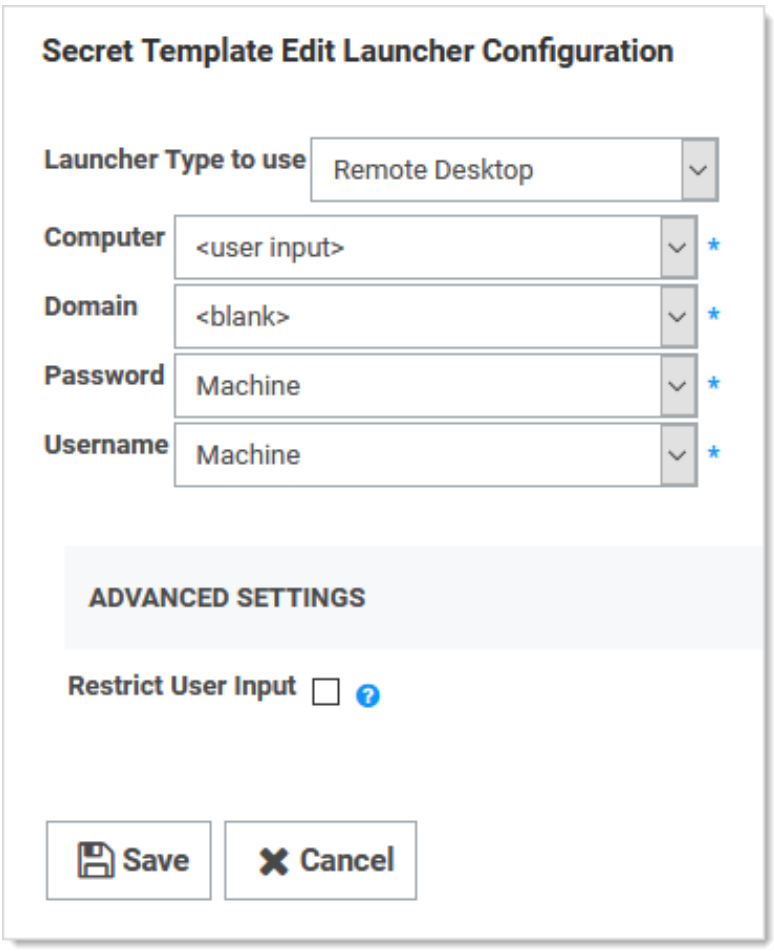
2. Click the unlabeled dropdown to select a secret template that you want to allow SSSC to launch from.
3. Click the **Edit** button to view that secret template. The Secret Template Designer page appears:



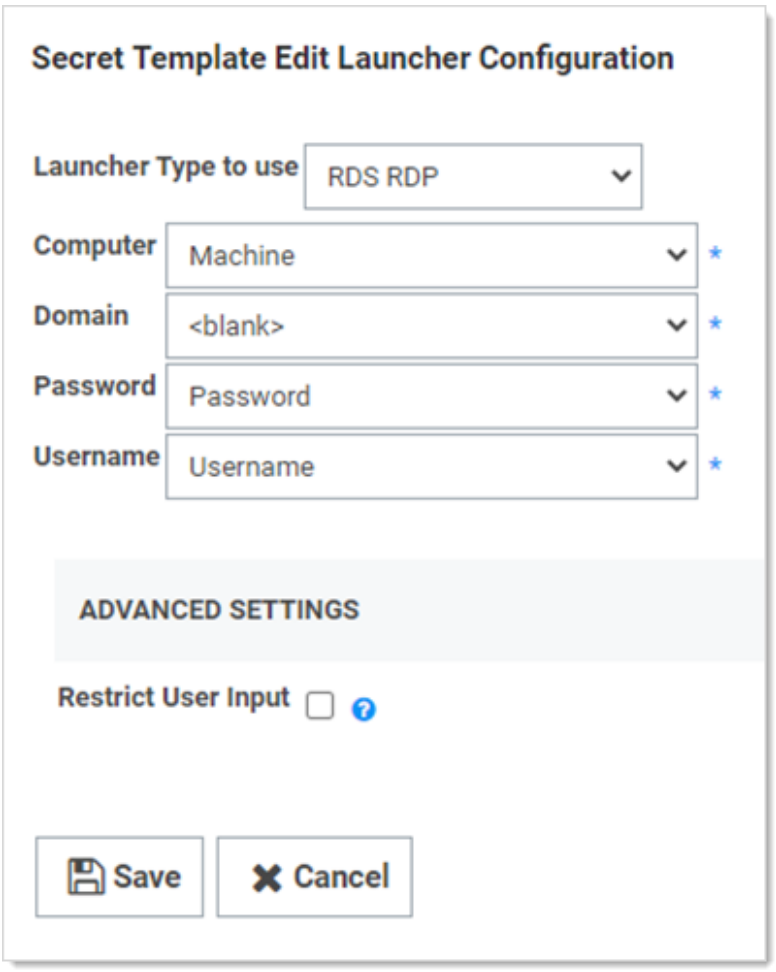
4. Click the **Configure Launchers** button at the bottom of the page to open the launcher mappings. The Secret Template Edit Launcher Configuration page appears:



5. Click the **Add New Launcher** button. A page of the same name appears:



6. Click to select the desired parameters for the launcher:



7. All secrets using this template are now ready to run SSSC launches.

Troubleshooting Session Connector

When launching a downloaded .RDP file, if SSSC rejects the session due to any issues (including being expired based on the "Session Connector Session Timeout" setting), the user's Remote Desktop client will receive a generic error about the RemoteApp being invalid.

In the `SS.log` file, you can search for "SessionConnector" to find details about why sessions may have been rejected.

Session Connector will also log to the file `C:\Program Files\Delinea Software Ltd\Secret Server Session Connector\log\SS-SC.log` on each RDS server. That is, If the RDS server has trouble using the supplied RDS credential to create a local user, it is logged to this file

Uninstalling Session Connector

Secret Server Session Connector can be removed from "Add/Remove Programs" or "Apps & Features." Once uninstalled, a reboot is required to restore the default Remote Desktop behavior.

Any related SSSC custom launchers need to be un-associated with any secret templates they were previously tied to.

> It is not currently possible to delete a custom launcher in Secret Server, but if it is unassociated with all secret templates, it will not appear on any secrets.

Web Launchers

Web launchers are a separate login method from the Web password filler and provide a convenient click to automatically log on simpler websites. Web launchers do not work on complex login pages that rely on JavaScript. For those login pages, use the browser extension for the Web password filler. By default, Web launchers are enabled on the Web Password Secret template, but they can be enabled on custom templates as well, as described in "Enabling Launchers" on page 675.

Configuring Web Launchers for Secrets

Once enabled on the template, a Web launcher needs to be configured for the secret. Each website login is unique and requires the secret text-entry fields to be mapped to the form controls. For a new secret the Launcher icon appears and clicking on it takes the user to a configuration screen. The user can also view and access the configuration screen from the Launcher tab. Depending on whether other secrets with the same website have been configured, the user has different options.

> Configuring the Secret for use with the Web Launcher requires the user to have Owner permission on the Secret.

First, there is the option of downloading the setting from Delinea.com. When the Configure Web Launcher page is loaded, Secret Server checks online at Delinea.com for pre-approved matching websites. If any are found, they are downloaded and made available to pick from in the dropdown list.

> This functionality can be disabled in Secret Server in the Configuration Settings.

The list displays all downloaded configurations and other secrets' configuration for the same domain that the user has permission to view. Select one from the list and click **Next** to create a copy of the settings for the secret.

There is also an option to create a configuration that allows the Web launcher to be used on most websites and not rely on published configuration settings. To use this, select the last item in the dropdown list and click **Next**. The next section discusses the create process.

Creating a Configuration

When configuring the Web Launcher:

- **Entering the Login URL:**Secret Server needs to know the exact URL used to login to be able to figure out the controls and perform the automatic login. Some example login URLs:
  - `https://login.yahoo.com/config/login`

- `https://MyServer/Billing/login.aspx`
- `https://firewall07/login/`

> The Login URL is typically a secure site with a prefix of `https://`. If allowed to access the site, Secret Server automatically detects if https should be used to ensure the credentials are passed securely.

- **Providing the Page Source:** If Secret Server is not allowed access to sites, or the login URL is not accessible by an external site, the page source needs to be provided for the Web launcher controls to be obtained. Ensure the login URL is correct when the page source is taken. If the site can be accessed by Secret Server the page source is automatically obtained and this step is not present.
- **Choosing the Form:** The page is read, and the exact login form needs to be identified. The page forms are listed in the list with the most likely selected. If no forms or no likely forms are found, the user needs to update the URL or page source, as configuration must have at least one textbox and one password box.
- **Wiring Up the Fields to Controls:** In most cases, Secret Server automatically wires up the Username and Password text fields to the correct page controls. If not, the user completes the control mapping on the Launcher tab.

Launching to a Website

The Web launcher can be used by clicking the Launcher icon on the Secret View page. The Web launcher opens a new window in the browser, which attempts to login to the site using the credentials on the secret. The uploaded configuration is reviewed and published by Delinea for all Secret Server customers to use with the check online feature. No secret or identifiable information is uploaded to Delinea.com. Only the website URL and control names are sent.

Installing Browser Extensions

Web Password Filler

To use the Web Password Filler, install one of the supported browser extensions as described below:

- Chrome: Install the extension by clicking the Web launcher icon in a Web Password secret, or by downloading it from the Google Chrome Web Store. Also see Managing Extensions in Your Enterprise on securely managing Chrome extensions at scale.
- Edge Chromium: Install the extension by downloading it from the Microsoft Edge add-ons site.
- Firefox: Install the extension by clicking the Web launcher icon in a Web Password secret, or by downloading it from the Firefox add-ons site.
- Safari: Install the extension by downloading it from Apple App Store. See the notes in the list below:
  - WPF supports Safari running on macOS Monterey and Big Sur 11.1.0, 11.2.1 and later.
  - WPF does not support the Native Messaging Host configuration in Safari browsers.
  - The Safari browser extension does not support Windows Admin Center.

- Session recording is not supported on Safari 15 and above.


Opera is not supported.

URL Lists

Web Password Filler supports URL lists. Users can map a list to the URL field in web launcher mappings, which can be useful in predefining various list categories of URLs that can then be used as a field in a secret to allow a secret to be tied to various sites. This is especially convenient for dynamic secrets, such as Active Directory. For more information on creating URL lists and launching comma-separated URLs, please refer to the Launching Comma-Separated URLs documentation.

Protocol Handlers

A *protocol handler* is an application on an end-user's machine. It enables communication between Secret Server and that client machine. It also provides the files needed by launchers. When a Secret Server user starts a launcher:

1. The protocol handler bootstraps the client-side application.
2. The protocol handler communicates with Secret Server over HTTP(S) to ensure that it is the latest version. If not, it begins an upgrade process.
3. The protocol handler bootstraps the target launcher type and begin the process of securely logging in the user. Beyond HTTP(S) transport protection, credentials are retrieved securely from Secret Server using signed AES-256-encrypted messages.

Installing Protocol Handler Through Group Policy

Group policy allows you to install Secret Server Protocol Handler on specific computers and groups of computers in your domain. Installing through group policy does not require changes to your firewall.

Step 1: Prerequisites

The Secret Server Protocol Handler Installer requires that .NET Framework 4.8 or greater be installed on the client machine. Most machines should already have this installed.

Step 2: Downloading the MSI From Secret Server

1. Log in to Secret Server
2. Go to **Tools > Launcher Tools** and click **Download Protocol Handler** to download the MSI.

Step 3: Setting up a Network Share

1. Place the downloaded MSI file into a Network Share on your domain controller.
2. Give domain users read access to the share.

Step 4: Creating a Group Policy That Allows for the Installation of the MSI

1. Open up the group policy management console (**Start** >**AdministrativeTools** >**Group Policy Management**)
2. Expand the Forest and Domain nodes until you locate the domain on which you are installing Secret Server Protocol Handler
3. Right click on Group Policy Objects and click **New**
4. Enter a descriptive name for your GPO (such as Secret Server Protocol Handler Installation) and click “OK”
5. Right click on the newly created GPO node and click **Edit**.
6. Select **Computer Configuration** > **Policies** > **Software Settings** > **Software Installation**
7. Right click on the “Software Installation” node and select “New > Package”
8. Browse to the MSI on your network share (that is, `\\ServerMachineName\Shared` is a valid network share, while `C:\Shared` is not) and click **Open**.
9. Select the **Advanced** radio button and click **OK**.

   If you wish to have Secret Server Protocol Handler uninstalled when it falls out of the scope of management, then click on the “Deployment” tab and check the “Uninstall this application when it falls out of the scope of management”

10. Click **OK**
11. In the group policy management object editor, expand **Computer Configuration > Administrative Templates > System** and click on the Logon node
12. Right click on the “Always wait for the network at computer start-up and logon”, select **Edit**, click **Enabled**, and click **OK**.

Step 5: Linking Your Group Policy Object to an OU

If you want to install Secret Server Protocol Handler for specific computers and not for an entire OU, then the MSI allows for manual installation directly

1. Open up the group policy management console (**Start > Administrative Tools > Group Policy Management**)
2. Expand the Forest and Domain nodes until you locate the domain on which you are installing Secret Server Protocol Handler
3. To link the GPO to an entire OU:
   - Right-click the Organizational Unit for which you want Secret Server Protocol Handler to be installed and select **Link an Existing GPO**.
   - Select the GPO you created in "Step 4: Creating a Group Policy That Allows for the Installation of the MSI" above above.
   - Click **OK**

The OU is now linked to the GPO. To immediately force the group policy change and install the software on a client machine, open a command console on the client machine (start > run > cmd), type `gpupdate /force`, and restart the client machine. You can also wait for the group policy to go into effect, which usually takes one to two hours.

Step 6: Verifying the Configuration

1. **Start** > **Administrative Tools** > **Active Directory Users and Computers**
2. Right-click the Organizational Unit for which Secret Server Protocol Handler is now configured and select **All Tasks > Resultant Set of Policy**.
3. Check the box next to **Skip to the final page of this wizard without collecting additional information** , then click **Next** and **Next** again.
4. Click **Finish.**
5. In the new “Resultant Set of Policy” window, expand **Software Settings** under **Computer Configuration** and select **Software installation**.
6. “Secret Server Protocol Handler” should be visible under the **Installed Applications** column.

Managing Multiple Secret Server Instances with Protocol Handlers and Launchers

In an organization running multiple instances of Secret Server, some users might find themselves having to repeatedly uninstall and reinstall different versions of the protocol handler to match the different instances of Secret Server.

To enable Secret Server to simultaneously support multiple versions of the protocol handler, you just need to disable the protocol handler auto-update function using the procedure below.

Disabling auto-update for forward and backward compatibility is supported on the protocol handler only. The ClickOnce launcher and the Mac protocol handler do not support disabling auto-update.

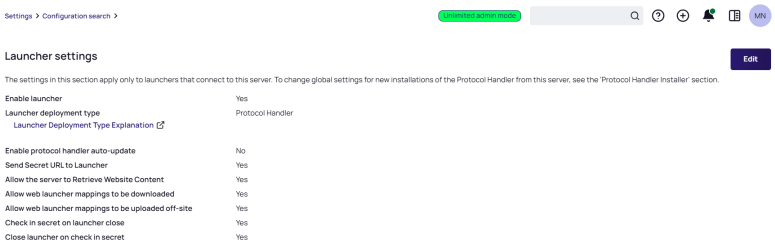
Prerequisites

- Secret Server Cloud 10.8+
- Secret Server On-Premises 10.8+
- Protocol Handler 6.0.0.23 or higher on your PC

Setup Steps and Configuration

1. In Secret Server, navigate to **Admin > Launcher Settings**, and click **Edit**.
2. Uncheck the box next to **Enable Protocol Handler Auto-Update**, and click **Save**.

3. Ensure that the **Enable Protocol Handler Auto-Update** function is now labeled, **No**.



You can re-enable protocol handler auto-update at any time by following the steps above and re-checking the box next to **Enable Protocol Handler Auto-Update**. When you re-enable auto-update, users will be required to install the latest instance.

> While protocol handler auto-update is disabled, each user must manually update their installed protocol handler as necessary on a machine-by-machine basis. The MSI can be installed directly or through Group Policy. A reboot may be necessary on certain operating systems.

Manually Updating Protocol Handler

1. In Secret Server, navigate to **Admin > Launcher Tools**.
2. On the **Launcher Tools** page, in the **Protocol handler installer** section, click **Download Protocol Handler (64-bit)** to download the file.
3. Follow the steps in the installation wizard.

Protocol Handler Administrative Settings

The Secret Server protocol handler has several administrative settings that you can configure through Microsoft's Group Policy Objects (GPOs) or through Secret Server itself.

> We **strongly** recommend using GPOs instead of Secret Server.

Available Settings

Allowed Secret Server Domains

This setting controls which domains or IP addresses the protocol handler installation may connect to. If the setting is unset or disabled, then the protocol handler is allowed to connect to any domain. If one or more comma-separated values are provided, then the protocol handler is blocked from accessing any domains or IP addresses not included in the list.

The protocol handler performs a string match against the URL it receives. It does not attempt to resolve domain names to IP addresses. Values in this list should match only the domain or IP address portion of the actual URL used to access Secret Server. For example, if users access your installation via `https://example.com/SecretServer`, then `example.com` should be added to the list. If `example.com` resolves to the IP address `192.168.1.5`, then adding that IP address *will not* allow access to the domain if users actually access it via `example.com`.

Wildcards are not supported, but subdomains do matter. The above entry for `example.com` would not allow `www.example.com`; the two may need to be added separately depending on your configuration. Ports and protocols are also unnecessary—only the domain portion is checked. For example, do not include an entry in the list like `https://example.com` or `example.com:885` as both are invalid. Simply using `example.com` covers these scenarios.

Disable Auto-Update

This setting ensures protocol handler will never auto-update itself, even if told to by the Secret Server installation that it connects to. When the setting is enabled, protocol handler installations need to be updated either manually or as part of your organization's regular program-update process.

Configuration Methods

Choosing the Configuration Method

Important: If your domain is configured to use GPOs, we **strongly recommend** using that to configure the protocol handler.

Why use GPOs instead of Secret Server?

- GPOs are more resilient, as Windows reapplies settings if they are deleted from the registry. Settings applied through Secret Server have no such resilience.
- GPOs are centrally managed along with other settings for machines in your domain.
- For security reasons, Secret Server's configuration can only be applied during the initial installation of Secret Server. If you change these settings within Secret Server, users must reinstall the program before they will be applied. GPOs do not have this restriction.

Configuring GPOs

You can download GPO definitions for your version of Secret Server from the Launcher Tools page of the Admin section of Secret Server. For details about using these policy definition files, see How to create and manage the Central Store for Group Policy Administrative Templates in Windows. Both machine and user configurations are available as needed, but machine configurations will always override user configurations—if a machine configuration is configured, the user configuration is completely ignored.

Settings are available in the group policy editor under **(Computer/User) Configuration > Administrative Templates > Secret Server Protocol Handler**.

Configuring Settings During Secret Server Installation

If you do determine that using Secret Server's settings are necessary, you can configure them via the Configuration page in the Admin section of Secret Server, in the "Protocol Handler Settings (Install-Time)" section. Enabling these settings causes downloads to generate a zip rather than an MSI file. The zip file contains a batch file that configures the install-time settings. These settings only update when the protocol handler is manually reinstalled or updated—changing them later on through Secret Server has no effect on protocol handlers that are already installed on user machines.

Secret Server Mobile Apps Overview

Both Secret Server Mobile and Delinea Mobile provide secure access to secrets and support multi-factor authentication (MFA). However, they have distinct features and functions tailored to different use cases. Below is a comparison of the two mobile applications:

Secret Server Mobile

Primary Functionality

- Connects to a Secret Server instance to view, manage, and use secrets.
- Supports multi-factor authentication, biometric unlock, autofill, online and offline caching, and advanced secret workflows.

Key Features

- **Multi-Factor Authentication**: Supports DUO Push, DUO Phone call, Pin Code, and TOTP authenticators.
- **Biometric Authentication**: Supports fingerprint (Android and iOS) and facial recognition (iOS only).
- **Autofill**: Automatically populates username and password fields on specified websites or other mobile applications.
- **Online and Offline Caching**: Secure storage for secrets, with options for offline access.
- **Advanced Secret Workflows**: Includes checking in or out secrets, submitting access requests, and handling approvals and denials.

User Interface

- Similar to the Secret Server desktop interface, making it easy for users to navigate and find secrets.
- Home screen with options to view secrets in different ways (All, Favorites, Recent, Shared).
- Side navigation panel for quick access to Inbox, Folders, Cached items, Change Password, Settings, Feedback, and Logout.

For more details, refer to the Introduction to the Secret Server Mobile Application.

Delinea Mobile

Primary Functionality

- Provides MFA verification for the Delinea Platform and portable access to secrets managed in Secret Server.
- Supports multiple Secret Server tenants for users needing access to multiple instances.

Key Features

- **Multi-Factor Authentication**: Supports Delinea Mobile Authenticator and biometric unlock.
- **Biometric Authentication**: Supports fingerprint (Android and iOS) and facial recognition (iOS only).

- **Autofill**: Automatically populates username and password fields on specified websites or other mobile applications.
- **Offline Access to Secrets**: Allows users to download secrets and access them without network connectivity.
- **TOTP Capabilities**: Provides time-based one-time password (TOTP) capabilities for sites and web applications that support TOTP-based authentication.

User Interface

- Designed to provide a seamless user experience with easy navigation and access to secrets.
- Supports viewing, creating, and organizing secrets into folders.
- Includes features for setting up and managing MFA, viewing notifications, and handling offline access.

For more details, refer to the Delinea Mobile Overview.

Summary

- **Secret Server Mobile** is primarily focused on providing comprehensive access and management of secrets within the Secret Server environment, with robust support for secret workflows and offline caching.
- **Delinea Mobile** extends its functionality to include MFA verification for the Delinea Platform, supports multiple Secret Server tenants, and offers additional features like TOTP capabilities and offline access to secrets.

Both applications enhance security and productivity by enabling secure access to secrets and supporting various authentication methods, but they cater to slightly different use cases and user needs.

Setting Maximum Time for Secret Server Mobile Offline Caching

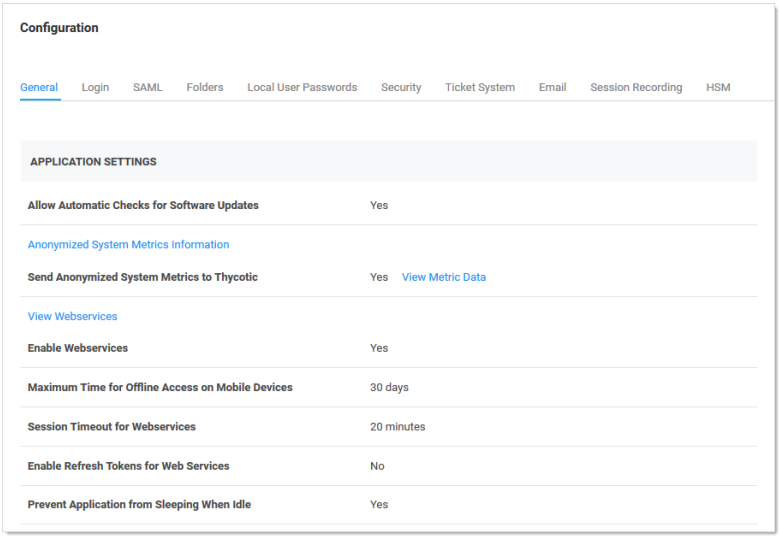
Overview

The **Maximum Time for Offline Access on Mobile Devices** setting in Secret Server determines how long to cache secret data on the mobile device. Once the device is not in contact with the server for longer than the specified amount of time, the device removes its cache of the stored secrets. The only way to view secrets on the device once the cache is cleared is to connect to Secret Server again so that the secrets can be re-downloaded and cached.

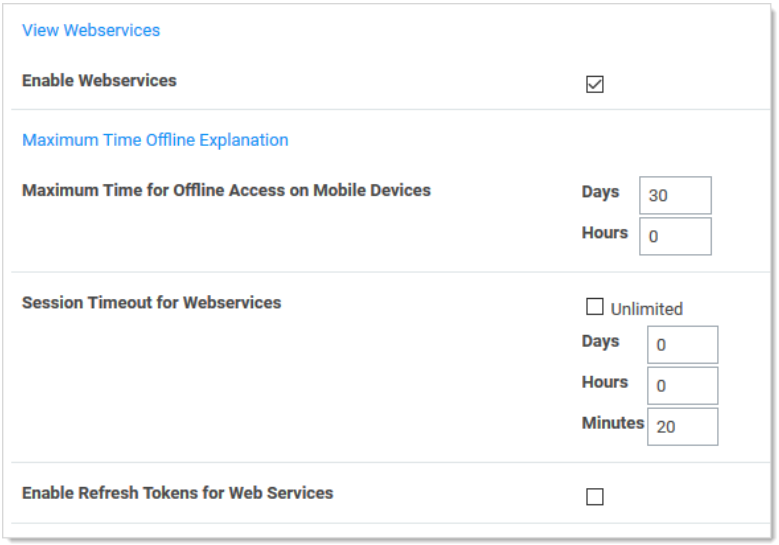
Procedure

To set the maximum time:

1. In Secret Server dashboard, click **Admin > Configuration**. The Edit Configuration page appears:



2. On the **General** tab, click the **Edit** button at the bottom of the page.
3. Click to select the **Enable Webservices** check box in the **Application Settings** section:



4. Type your preferred interval in the **Days** and **Hours** text boxes in the **Maximum Time for Offline Access on Mobile Devices** section.

> Setting the Maximum Time Offline to less then hour prevent the device from caching as the cache window is too small.

> Because caching all secrets creates an audit record in the database for each secret, we recommend not setting the window too short so that users constantly need to cache all secrets.

5. Click the **Save** button at the bottom of the page.

Example

An example of a cache window:

If Maximum Offline Time is set to 7 days, a iPhone user can cache secrets. If the iPhone has connectivity every hour the iPhone is used, it will check in with the server. Each time the iPhone checks in the 7 days, the cache window is extended. Thus, if the user uses the app once every 7 days, the app cache will remain. If the user does not have connectivity (such as in Airplane Mode) or does not turn on the app for longer than 7 days, then the next time the app is used the cache will be cleared because the maximum allowed time offline has been surpassed.

Secret Server Networking Overview

Secret Server offers a comprehensive suite of networking features designed to secure and manage privileged access. Below is an overview of key components including messaging, distributed engines, RDP proxy, HTTP, and SSH.

Messaging

Secret Server uses RabbitMQ as its message bus or broker to facilitate message traffic between various components. This asynchronous message-based system ensures that operational instructions and data are passed back and forth efficiently. All messages are encrypted during transit, and any accumulation of messages in a queue is considered abnormal, indicating a potential application problem.

Distributed Engines

Distributed engines in Secret Server provide scalability and rapid results for large networks. They use secure network communication, queueing, and parallel processing to manage tasks such as password changes and discovery. A distributed engine consists of site connectors, sites, and engines, which work together to distribute and process work items efficiently.

RDP Proxy

Secret Server uses an SSH Proxy to secure RDP connections within a Privileged Access Management (PAM) solution. This approach mitigates the risks associated with direct RDP access by routing all RDP and SSH connections through the Secret Server host or a Distributed Engine. This setup ensures that only authorized sessions can access target hosts, and it prevents malware from spreading laterally between machines.

HTTPS

You can configure Secret Server to run with multiple front-end web servers, offering clustering for redundancy and load balancing. This setup ensures better performance and limits potential downtime from a single point of failure.

The backbone bus, typically RabbitMQ, handles all internal communication between roles in a clustered environment.

SSH

Secret Server's SSH Proxy routes SSH sessions to protect endpoint credentials. It can be configured to proxy through the Secret Server web application or a distributed engine. This setup allows for secure and monitored SSH access, ensuring that all connections are authorized and audited.

These components work together to provide a secure, scalable, and efficient networking environment for managing privileged access in Secret Server.

Secret Server Architecture

Architecture diagrams are contained in the Architecture section of the Delinea documentation.

Proxied Environments

If your Secret Server has outbound access through a proxy, its web.config must be modified to specify the proxy configuration.

- If Secret Server is also clustered and has multiple worker roles enabled, the web.config must be updated for each Secret Server in the cluster.

Microsoft has more information on this.

> **Note:** For more information about using a distributed engine through a proxy, please refer to "Distributed Engine Overview" on page 723. The other option in a clustered environment is to specify a remote site for the data upload, and upload data through a Distributed Engine. If the distributed engine's host server is also behind a proxy, however, the engine's Delinea Delinea.DistributedEngine.Service.exe.config must be modified similarly to the web.config in order to specify the proxy settings.

- For Secret Server v10.4 or later, the web-proxy.config can be uncommented and updated to specify the proxy settings.

  For Secret Server v10.3.000015 or earlier, you must add proxy-related XML to the web.config file immediately following the file's closing \</configSections\> tag, as depicted here:

```
</configsections>
    <system.net>
        <defaultproxy enabled="true" usedefaultcredentials="true">
             <proxy
usesystemdefault="false" proxyaddress="https://proxy:port" bypassonlocal="true"/>
        </defaultproxy>
    </system.net>
<configuration type="thycotic.foundation.configuration, thycotic.foundation">
```

Using Webnode with Proxied Environments

If you are using a webnode, you will need to add the following code:

```
<system.net>
    <defaultProxy configSource="web-proxy.config" />
</system.net>
```

Example:

```
59    <!-- Please see the file "web-appSettings.config" to change appSetting
60    <appSettings file="web-appSettings.config"/>
61    <!-- Please see the file "web-log4net.config" to change general logging
62    <log4net configSource="web-log4net.config" />
63      <system.net>
64          <defaultProxy configSource="web-proxy.config" />
65      </system.net>
66    <runtime>
67      <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
```

Distributed Engine (DE) Configuration

Note: Delinea suggests that you review any exception in the proxy configuration for both web nodes and DEs as a bypass. The configuration files may be overwritten with product updates and changes will be need to be reviewed, and possibly implemented again.

When Secret Server and DEs are behind a proxy, certain settings need to be added to webnodes and DEs if they exist in the environment.

To use with the Distributed Engine through a proxy, you will need to add proxy info to Delinea.DistributedEngine.Service.exe.config between </system.serviceModel> and located in the C:\Program Files\Thycotic Software Ltd\Distributed Engine\ folder on the distributed engine. You may need to refer to this article for other proxy related settings:

Element (Network Settings)

```
<system.net>
    <defaultProxy>
        <proxy usesystemdefault="true" />
    </defaultProxy>
</system.net>
```

```
  </system.serviceModel>
  <system.net>
    <defaultProxy
                enabled = "true"
                useDefaultCredentials = "true">
      <proxy autoDetect="false" bypassonlocal="false" proxyaddress="http://127.0.0.1:8888" usesystemdefault="false" />
    </defaultProxy>
  </system.net>
</configuration>
```

Important: You will need to restart the DE service after this update and the setting will need to be reapplied after any DE upgrade.

Webnode Configuration

Main Proxy settings are stored in the web-proxy.config file in the Secret Server folder on each webnode. Microsoft's article on Proxy configuration explains all settings.

Element (Network Settings)

Example #1

```
<?xml version="1.0" encoding="utf-8" ?>
<defaultProxy enabled="true">
<proxy
    usesystemdefault="true"
    proxyaddress="http://192.168.1.1:8080"
    bypassonlocal="true"
  />
</defaultProxy>
```

Example #2

```
<defaultProxy enabled="true">
    <proxy proxyaddress–"http://proxy.domain.com:80" bypassonlocal-"true" / >
</defaultProxy>
```

Now, the following files need to be edited to point to the web-proxy.config file.

- web-embeddedRole-backgroundScheduler.config
- web-embeddedRole-backgroundWorker.config
- web-embeddedRole-engineWorker.config
- web-embeddedRole-messageBroker.config
- web-embeddedRole-sessionRecordingWorker.config

The code used in these files can be as follows:

```
<system.net>
    <defaultProxy configSource="web-proxy.config" />
</system.net>
```

```
<!-- Please see the file "web-appSettings.config" to change appSetting configuration. -->
<appSettings file="web-appSettings.config"/>
<!-- Please see the file "web-log4net.config" to change general logging configuration. Cha
<log4net configSource="web-log4net.config" />
  <system.net>
  <defaultProxy configSource="web-proxy.config" />
</system.net>
<runtime>
```

Note: Placement of this setting may effect connection. Make sure this setting is placed before the <runtime> element has been confirmed to make the configuration work.

Troubleshooting Tips

Distributed Engines

Check the SSDE.log file.

Example:

```
2022-09-20 15:19:49,756 [CID:] [C:] [TID:PriorityScheduler Elastic Thread @ AboveNormal] ERROR Thycotic.SecurityAnalytics.DataUploader.Clients.Aws.SQSClient - Failed enqueuing
object to queue  - (null)
Amazon.Runtime.AmazonServiceException: A WebException with status ConnectFailure was thrown. ---> System.Net.WebException: Unable to connect to the remote server --->
System.Net.Sockets.SocketException: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed
because connected host has failed to respond 3.236.169.124:443
   at System.Net.Sockets.Socket.DoConnect(EndPoint endPointSnapshot, SocketAddress socketAddress)
   at System.Net.ServicePoint.ConnectSocketInternal(Boolean connectFailure, Socket s4, Socket s6, Socket& socket, IPAddress& address, ConnectSocketState state, IAsyncResult
asyncResult, Exception& exception)
   --- End of inner exception stack trace ---
   at System.Net.HttpWebRequest.GetRequestStream(TransportContext& context)
```

Webnodes

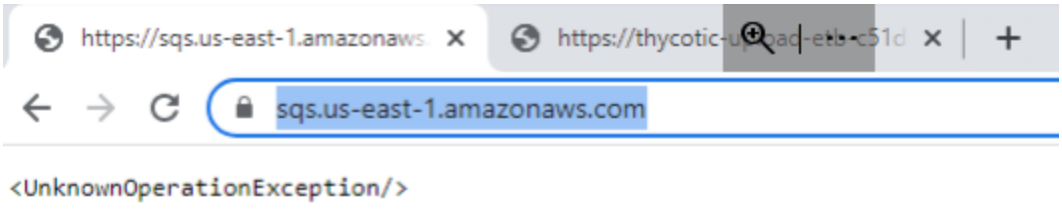
Check the SS-BWSR.log file

Example:

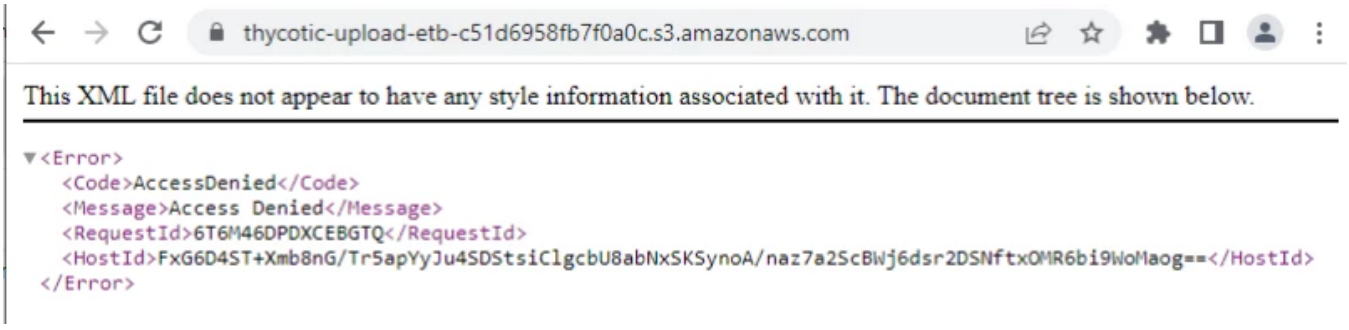
```
2022-08-05 21:21:25,647 [CID:c14908f64c834eb79d5b67c21430b1bb] [C:] [TID:PriorityScheduler Thread @ BelowNormal] DEBUG Thycotic.AppCore.Logging.SystemLogger - Publishing
Log Message - (Amazon.Runtime.AmazonServiceException: A WebException with status NameResolutionFailure was thrown. ---> System.Net.WebException: The remote name could not
be resolved: 'thycotic-upload-etb-c51d6958fb7f0a0c.s3.amazonaws.com'
   at System.Net.HttpWebRequest.EndGetRequestStream(IAsyncResult asyncResult, TransportContext& context)
   at System.Net.HttpWebRequest.EndGetRequestStream(IAsyncResult asyncResult)
```

Check to see if the machine can pull up the IP/URL in the logs in web browsers. While these do not specifically show full communication between systems, they can help narrow down the issue.

Example #1:



Example #2:



Distributed Engine Overview

> Distributed engine configurations for Secret Server On-Premises and Secret Server Cloud are not 100% equivalent. To view the differences, please refer to our reference architectures for Secret Server. They are, at minimum, refreshed every year and are created by our Professional Services Solutions Architect team.

> Distributed engine upgrades are no longer mandatory for every release. The Distributed Engine Configuration page can set the minimum required engine version. Modifying this triggers an automatic update for any engine below this version.

In the action menu for an engine on the Sites page, you can start a manual upgrade for individual engines below the latest version, which prompts the engine to update when it next calls in.

When changes are made needing an upgrade, the minimum required version is updated during the update process, and all engines update immediately.

Overview

Out of the box, Secret Server performs all functions from the Web server it is installed on; however, specific features can be routed through a distributed engine for enhanced performance. For example, synchronize and authenticate AD users can be done in Secret Server via your local site or from a distributed engine (DE).

You can install a DE in a remote site and allow it to operate many functions. Communication with Secret Server Cloud also requires the distributed engine to be installed. Note that you should not install a DE on the same machine running Secret Server.

Architecture and Workflow

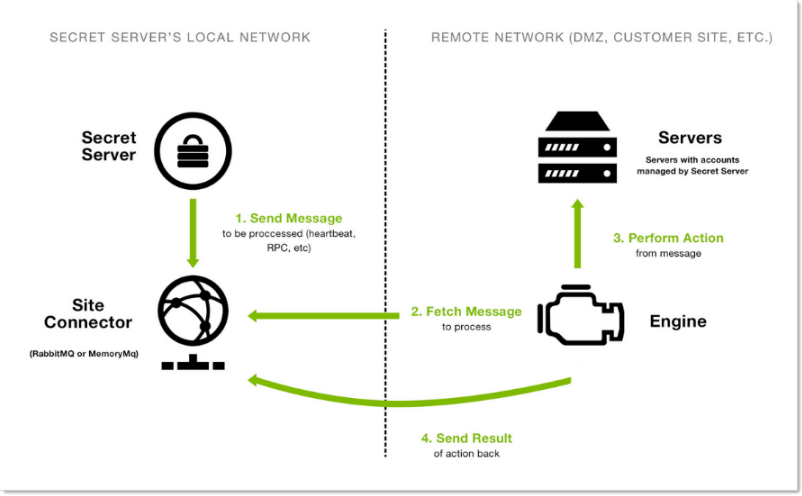
Main Components

DEs support heartbeat, Remote Password Changing (RPC), and discovery. A DE is composed of site connectors, sites, and engines:

- A **distributedengine** is a Windows service that does the actual work, such as password changing, heartbeat, discovery, and more. Each engine belongs to a site.
- A **site** can be thought of as a bucket of work items for a particular network area. Each engine is assigned to a single site, but each site can include multiple engines, significantly increasing throughput.
- A **site connector** is a Windows service that holds the work items for a number of sites. The site connector can be either RabbitMQ or MemoryMQ (a built-in service developed by Delinea). Each site can only be assigned to a single site connector, but you can have multiple site connectors running on separate machines, each storing work items for multiple sites. Those sites, in turn, distribute the work items among multiple engines. The ability to add new Site Connectors, Sites, and Engines as needed makes Distributed Engine a highly-scalable solution.

For the highest scalability and reliability, Delinea recommends using RabbitMQ. MemoryMQ is an easier but less capable alternative and can be used for trials and proof of concepts but **should not** be used for production environments. Two exceptions are very small deployments and customers that do not use open-source software for compliance reasons.

Figure: Distributed Engine Components



The above diagram is a simplified, conceptual one, not a network diagram. It does not show a callback port from the DE to Secret Server. DEs require either an HTTPS or TCP port to communicate with Secret Server for initial activation, updates, and continuous periodic check of site and site connector settings.

Ports

DEs have two configurable ports: one for connecting to the site connector, and one for the engine to retrieve configuration information from Secret Server at regular intervals. The callback port from an engine to Secret Server can be configured to contact the website directly over HTTP, HTTPS, or TCP. HTTP and HTTPS connections use the existing IIS port bindings. All connections are outbound—no inbound connections are made from Secret Server or the site connectors to the remote networks.

Default ports:

- RabbitMQ: 5672 (non-SSL), 5671 (SSL)
- MemoryMQ: 8672 (non-SSL), 8671 (SSL)
- Secret Server: existing IP address bindings or custom port over TCP. We reserve one port for legacy upgrades, usually port 9999.
- Secret Server Cloud:
  - 443 (Web sockets—the default)
  - 5671 and 5672 (AMQP)

These ports are used for outbound traffic for engines to communicate with Secret Server Cloud instances. They are set by the "Azure ServiceBus Transport Type" global engine setting.

Security

Distributed engines have multiple security layers:

- Engines must be approved within Secret Server before they will be given access to a site.
- Work items are encrypted with a site-specific symmetric key prior to sending them to the site connector.
- Communication to the site connector supports SSL and TLS.
- Direct communication from engine to Secret Server uses a public-private key exchange.
- The engine configuration file is DPAPI encrypted.

For more information about DE security, see the "Distributed Engine Hardening" on page 1371.

Engine Workflow

This workflow applies to Secret Server On-Premises.

When an engine Windows service starts, the following steps occur:

1. The service contacts Secret Server directly using the engine callback port.
2. The service receives configuration information for the site connector to connect to and what site to process work items for.
3. The service connects to the site connector and registers with the site for work item processing.
4. The service fetches a work item from the site.
5. The service processes the work item.
6. The service gives the site the result of the processing, such as heartbeat success or discovery results.
7. The service fetches another work item, and the process continues.

Configuring Distributed Engines

When DEs auto update, they remove the MySQL, Oracle, and other DLLs that were manually placed there. To forestall this, we recommend creating an ignore file for DE upgrades. Please see How to create an ignore file for Distributed Engine upgrades for details.

Configuration

Below is a summary of the steps required to configure DEs:

1. Enable the DE and specify the engine callback settings.
2. Configure and Install the site connector.

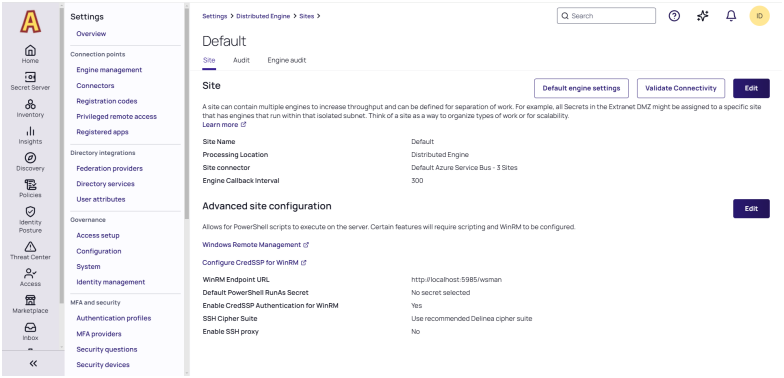
- If you plan to use RabbitMQ (recommended), please see "Installing RabbitMQ" on page 93. You can find general information on using RabbitMQ Helper to install RabbitMQ can be found in Installing Rabbit MQ.
- If you plan to use MemoryMQ, create the site connector record within Secret Server then click the **Download Site Connector Installer** button to get the MSI. Run the MSI on the desired host.

3. Setup sites.
4. Configure default DE settings. See "Engine Settings" below.
5. Install engines.
6. Assign secrets to sites. Secrets can be assigned to a site through their Remote Password Changing tab or via a bulk operation on the Secret Server dashboard. Once assigned to a site, all heartbeat or password changing operations take place through that site.
7. Assign discovery sources to sites. To run discovery through a site, edit the discovery source and assign the site. Once assigned, all discovery operations for that discovery source take place through that site.

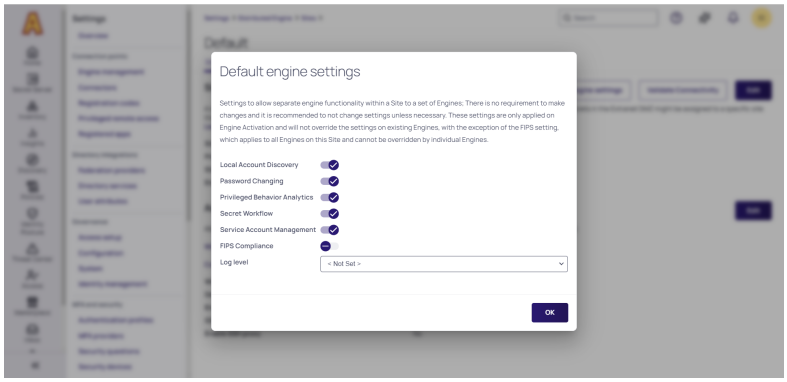
> Discovery should (a) have a dedicated DE and/or (b) a dedicated Site.

Engine Settings

1. If you click a site name, there is a **Default Engine Settings** button that allows you to pick which features of apply to new DEs connected to this Site:



2. Click the **Default Engine Settings** button.



3. When a new DE is installed, it inherits these settings from the site it is attached to. You can further customize engines by going to **Show Settings** in the Kebab menu for the engine.
4. FIPS Compliance is the only setting only available at the site level, and all DEs on the site use the same value. You cannot have a mix FIPS and non-FIPS engines on a single site. If you toggle the FIPS Compliance setting, Windows OS FIPS setting needs to be set to match it, and the engine server should be rebooted. The next time an engine checks in, if the engine's settings have been changed from those set on the site, the engine reloads itself and starts only performing the site's selected duties.

FAQ

What happens if Secret Server sends work items to the site connector, but no engines are running to consume them?

Work items continue to build up in the site connector until a limit is reached. Heartbeat work items have a Time To Live (TTL) of 5 minutes, Password Changing work items have a TTL of 20 minutes. Expired work items are thrown away and will not be processed. Once a heartbeat or password changing work item is sent to the site connector, Secret Server will not send the same work item to the queue until 5 minutes after the TTL is up (10 and 25 minutes for heartbeat and password changing, respectively). This prevents multiple pending heartbeat or password changing work items for the same secret at the same time.

How many Sites can a Site Connector hold?

MemoryMQ supports up to 100. RabbitMQ supports up to 200.

Can I cluster Site Connectors?

RabbitMQ supports clustering, MemoryMQ does not.

Can I use both RabbitMQ and MemoryMQ?

Yes. You can have as many site connectors, of either type, installed as needed. Note that while you can have both RabbitMQ and MemoryMQ installed on a single machine, you cannot have two RabbitMQ instances or two MemoryMQ instances on the same machine.

Can I convert a site connector from MemoryMQ to RabbitMQ or vice versa?

Yes. You can install the new site connector, swap the sites over to the new service, and then decommission the old site connector.

I've reinstalled my Distributed engine, but why is it not showing in the console?

If you have distributed engines installed and encounter an error or decide to reinstall distributed engine on a machine that had it previously, please be aware that the Add/Remove Programs option does not clear out the installation folder of the Distributed Engine (C:\Program Files\Delinea Software Ltd\Distributed Engine\)

This leaves behind some configuration files. If these are left behind during an uninstall, then during a reinstall it may cause issues where the distributed engine will not appear in the console for activation or assignment to a site.

Distributed Engine Hardening

Introduction

This topic discusses best practices for hardening Secret Server distributed engine servers.

If attackers compromise one of the DE servers, they would have access to all critical DBs, applications, and network devices at the network level. DEs do not store any passwords, PII, or user data in any configuration files.

> Due to their intrinsic nature, some PowerShell script run by DEs may expose API usernames or passwords in the PowerShell log.

General Hardening Steps

Restrict RDP Connections

- Limit RDP connections to all PAM Server, except for PAM admins and some users from the hosting team.
- If there is no firewall segmentation in LAN network, you can accomplish this with the Windows OS firewall.

Restrict Incoming Port Access to All DE Servers

- Allow only RDP port access from some internal IPs.
- Allow a SSH proxy port coming from the user's LAN.
- Block all other incoming ports.

Remove Unnecessary User Groups

For administrator and Remote Desktop user groups:

- Remove default domain admins, administrator and some common groups.
- Create one group that is going to have access these servers.
- Disable the built-in local administrator user.

Rename Default Accounts

- Change the names of both the administrator and guest accounts to names that do not indicate their permissions.
- Create a new locked and unprivileged "administrator" user name as bait.

Disable Services

Disable these services:

- Routing and remote access
- Smart card
- Smart card removal policy
- SNMP trap
- Special administration console helper
- Windows error reporting service
- WinHTTP Web proxy auto-discovery service

Restrict Network Protocols

Keep these:

- Client for Microsoft network
- File and printer sharing for Microsoft network
- Internet protocol version 4 (TCP/IPv4)

Remove these:

- QoS packet scheduler
- Link-layer topology discovery mapper IO driver
- Link-layer topology discovery responder

Validate Server Roles

Ensure only the minimum roles and features that are required are defined on the DE Servers. Remove all unnecessary roles and features.

The following roles are removal candidates, not ones to keep.

Roles

Application Server

- TCP port sharing
- Windows process activation service support
- Named pipe activation
- TCP activation

Remote Access

- Direct access and VPN (RAS)
- Routing
- Web application proxy (with dependent features)

Web Server (IIS)

- Web server
- Health and diagnostic
- Logging tools
- Tracing

Security

- Centralized SSL certificate support
- Client certificate mapping authentication
- Digest authentication
- IIS client certificate mapping authentication
- IP and domain restrictions
- URL authentication

Application Development

- Server side includes
- Web socket protocols
- Windows deployment services (with dependent features), including all child roles

Features

- Group policy management
- IIS hostable Web core
- Ink and handwriting services
- Media foundation
- RAS connection manager administration kit (CMAK)
- Remote server administration tools, including all child features.
- Windows internal database
- SMB 1.0/CIFS file sharing support

SSL/TLS Settings

Keep your server SSL/TLS settings up to date. Among other settings, the different protocols and cipher suites can be vulnerable to different attacks on SSL/TLS.

- Disable SSL 2.0
- Disable SSL 3.0
- Disable TLS 1.0
- Disable TLS 1.1
- Enable TLS 1.2

GPO Hardening

The following are recommended settings for Microsoft Group Policy Objects (GPO).

User Configuration > Policies > Administrative Templates > Control Panel/Personalization

Vulnerability:

There is no protection against a user with physical and remote desktop access to the server.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Enable screen saver	Enabled
Force specific screen saver	Enabled
Password protect the screen saver	Enabled
Screen saver timeout	Enabled Seconds: 600

Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies /Security Options

This setting enables advanced auditing in the operating system.

Policy	Recommended Value
Audit: Force audit policy subcategory settings to override audit policy category settings	Enabled

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Logon Account

Vulnerability:

Lack of information on unauthorized user login attempt. Lack of this type of information prevents identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Credential Validation	Success, Fail
Other Account Logon Event	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Account Management

Vulnerability:

Lack of information on user management in the system (addition and removal of users). Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Application Group Management	Success, Fail
Computer Account Management	Success, Fail
Distribution Group Management	Success, Fail
Other Account Management Events	Success, Fail
Security Group Management	Success, Fail
User Account Management	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Logon\Logoff

Vulnerability:

Lack of information on unauthorized user login attempt. Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Account Lockout	Success, Fail
Logoff	Success, Fail
Logon	Success, Fail
Network Policy Server	Success, Fail
Other Logon\Logoff Event	Success, Fail

Policy	Recommended Value
Special Logon	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Object Access

Vulnerability:

Lack of information on access to sensitive files and folders in the system. Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

Applying Auditing for Success can overload the system. In case an overload is created, it is recommended to apply the auditing for Failure only.

Policy	Recommended Value
Application Generated	Success, Fail
Certification Services	Success, Fail
Detailed File Share	Fail
File Share	Success, Fail
File System	Success, Fail
Kernel Object	Success, Fail
Registry	Success, Fail
Removable Storage	Success
SAM	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Policy Change

Vulnerability:

Lack of information on changes in the policy. Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Audit Policy Change	Success, Fail
Authentication Policy Change	Success, Fail
Authorization Policy Change	Success, Fail
Filtering Platform Policy Change	Success, Fail
MPSSVC Rule-Level Policy Change	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Privilege Use

Vulnerability:

Lack of information on the use of system authorizations. Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Non Sensitive Privilege Use	Success, Fail
Sensitive Privilege Use	Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > System

Vulnerability:

Lack of information on system start-up, shutdown and system changes. Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Other System Events	Success, Fail
Security State Change	Success, Fail
Security System Extension	Success, Fail
System Integrity	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Event Log

Vulnerability:

There is a risk that many log records will not be saved due to the file's size.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Maximum application log size	100032 KB
Maximum security log size	100032 KB
Maximum system log size	100032 KB
Prevent local guests group from accessing application log	Enabled
Prevent local guests group from accessing security log	Enabled
Prevent local guests group from accessing system log	Enabled
Retention method for application log	As needed
Retention method for security log	As needed
Retention method for system log	As needed

Computer Configuration > Policies > Windows Settings > Security Settings > Registry

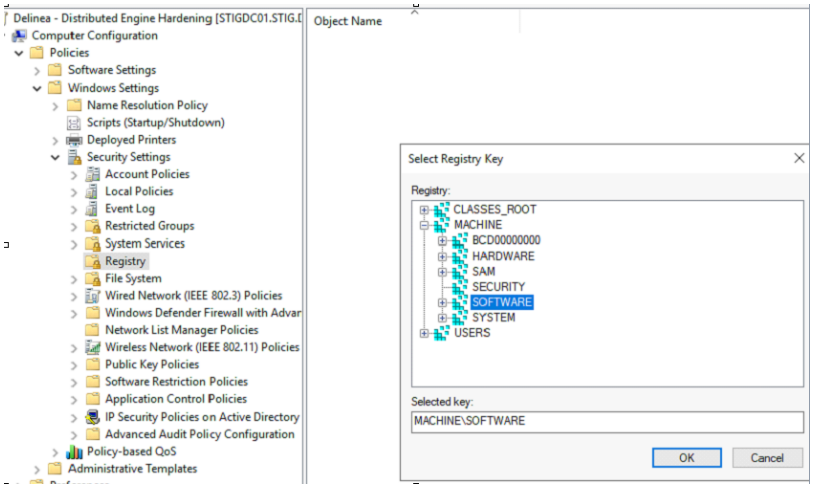
The purpose of this GPO setting is to add auditing to the following registry keys:

- `HKLM\SYSTEM`
- `HKLM\SOFTWARE`

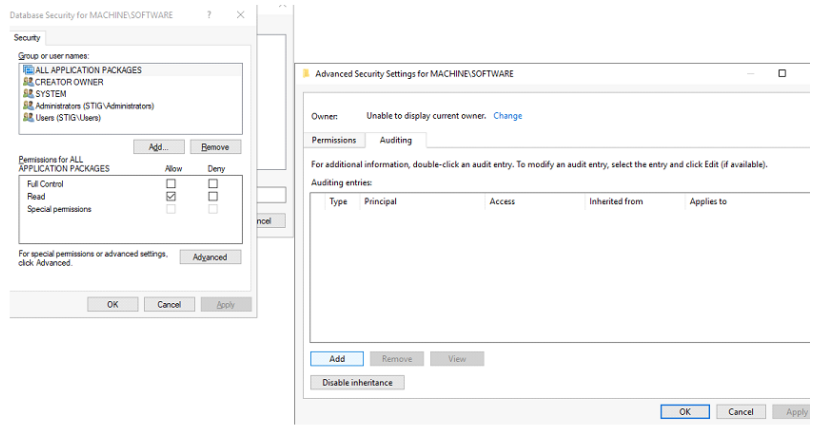
Auditing should be applied according to the following parameters:

- Audit - Success only: Set Value
- Audit - All: Create Subkey, Create Link, Delete, Read Permissions, Change Permissions

1. Right click on Registry, select Add Key, then select MACHINE\SOFTWARE.

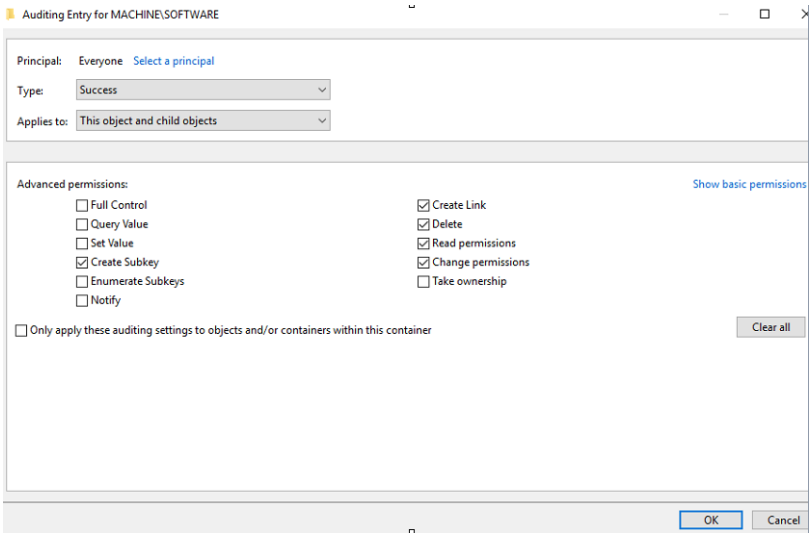


2. Click Advanced, select Auditing tab, click Add.

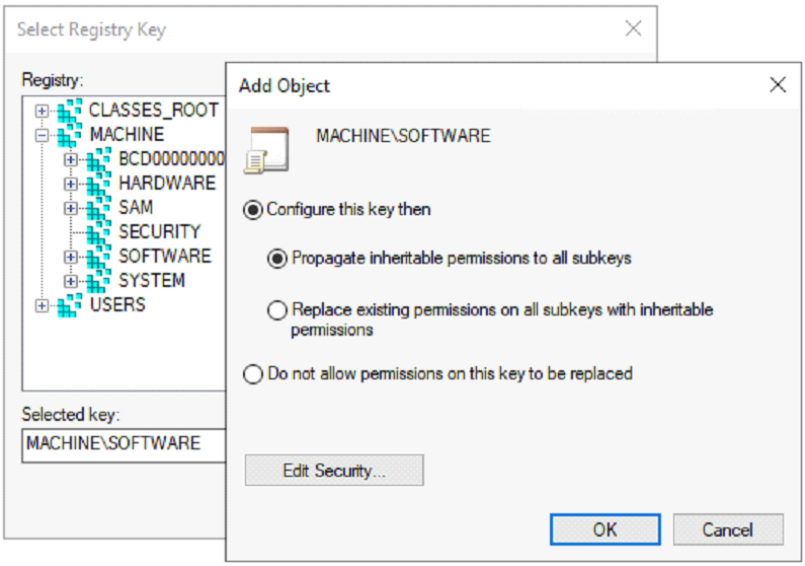


3. Change Principal to Everyone, select Show Advanced Permissions, select the following boxes:

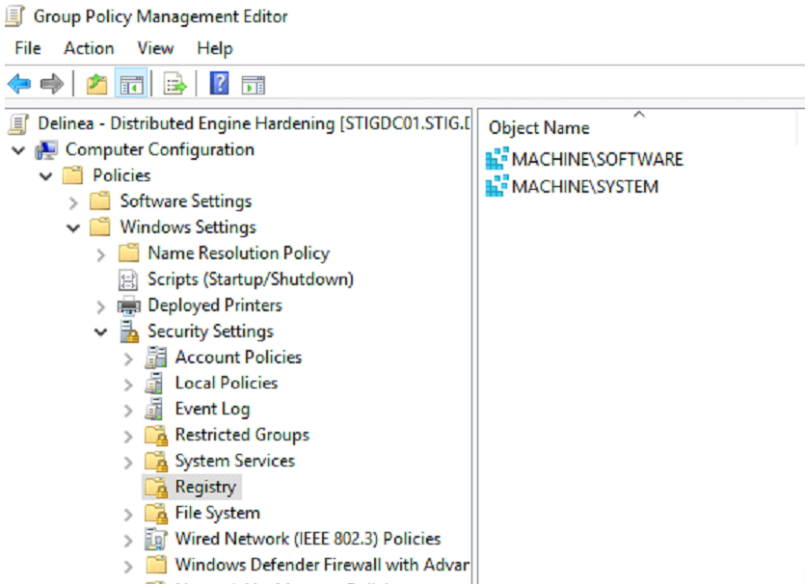
- Create Subkey
- Create Link
- Delete
- Read Permissions
- Change Permissions



4. Click OK, then click Apply Settings.



5. Perform the same steps above for MACHINE\SYSTEM.



Computer Configuration > Policies > Windows Settings > Security Settings > File System

The purpose of this GPO setting is to add auditing to the following directories:

- `%SystemRoot%\System32\Config`
- `%SystemRoot%\System32\Config \RegBack`

Vulnerability:

Lack of information on delete, change of authorizations, gain ownership of sensitive files, or any attempt to do so, will prevent the ability to identify unauthorized access and therefore will make it difficult to prevent such attempts.

Severity of the damage:

Medium

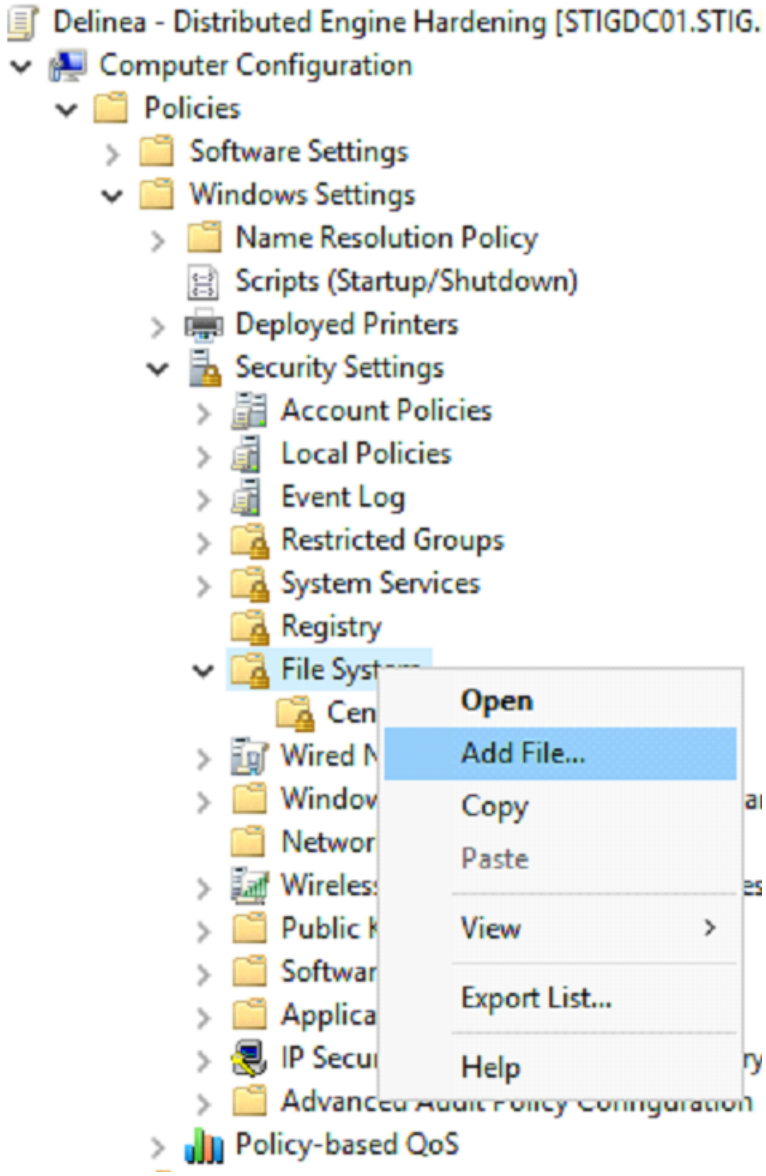
Operational aspects:

None

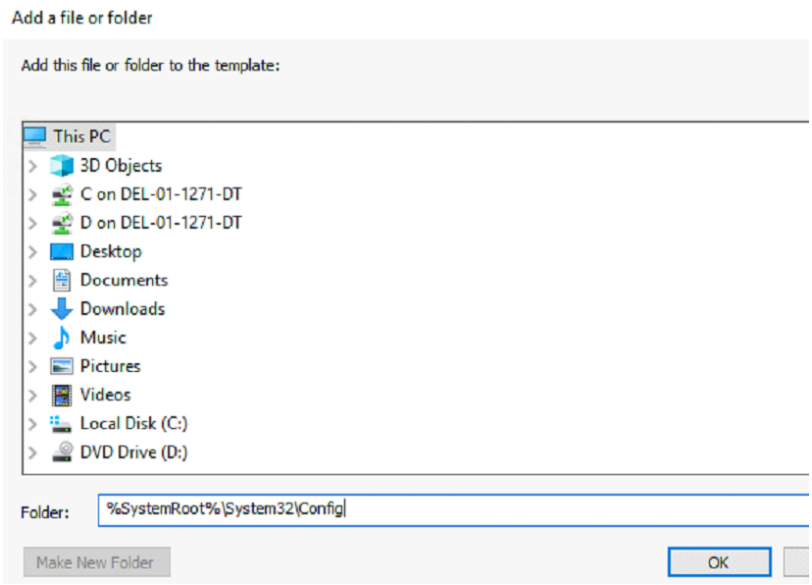
Permissions and auditing should be applied according to the following parameters:

- Audit-Failure only: Traverse Folder\ Execute File, List Folder\ Read Data, Read Attributes, Read Extended Attribute.
- Audit - All: Create Files\ Write Data, Create Folders\ Append Data, Write Attributes, Write Extended Attributes, Delete Subfolders And Files, Delete, Change Permissions, Take Ownership.
- Permissions: Administrator, System - Full

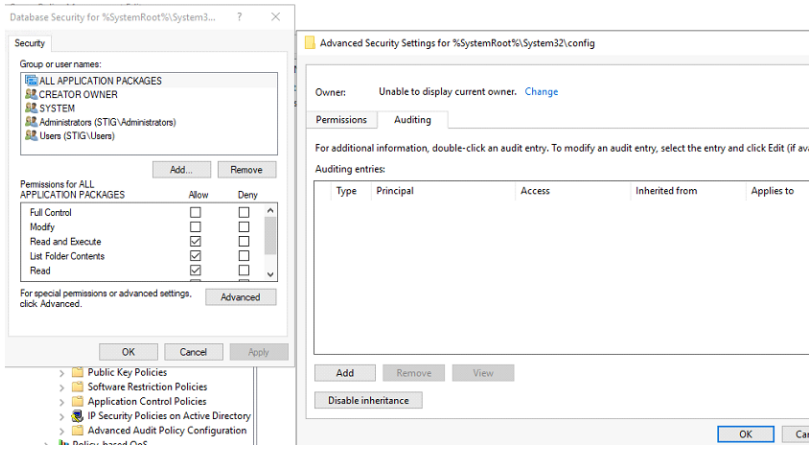
1. Right click File System, click Add File.



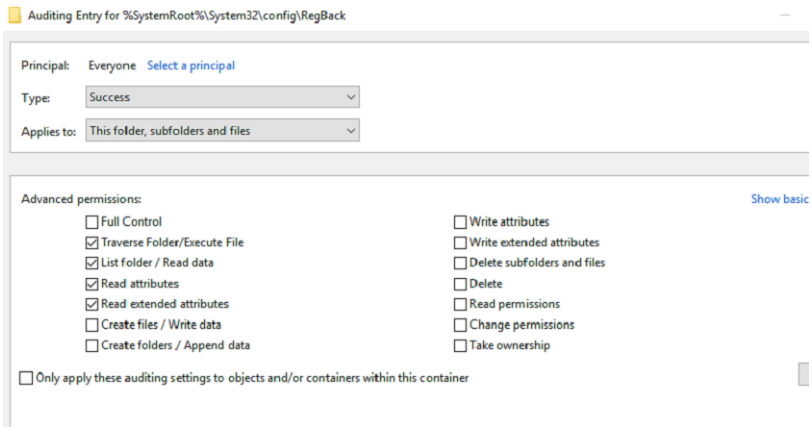
2. Add the folder path *%SystemRoot%\System32\Config*



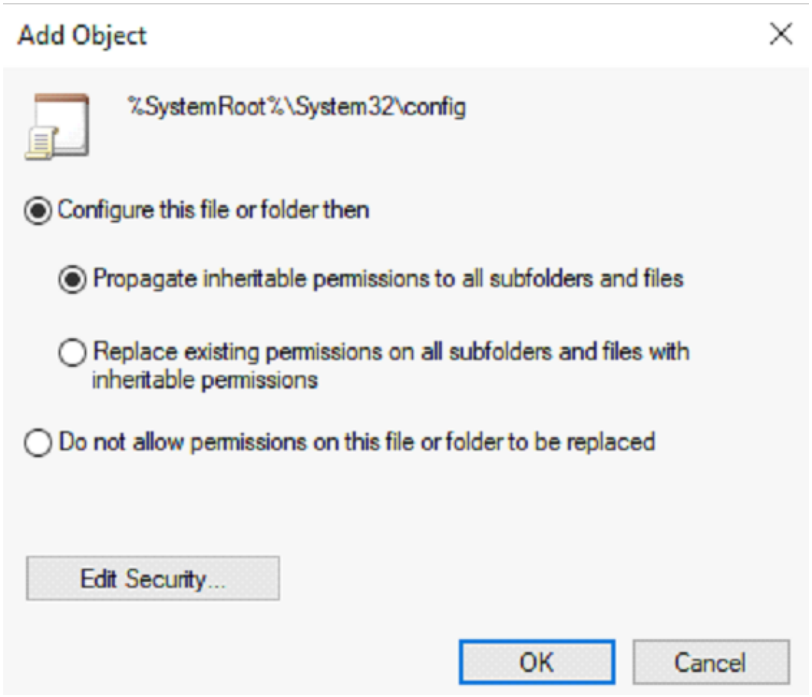
3. Click Advanced, then click Auditing tab, and click Add.



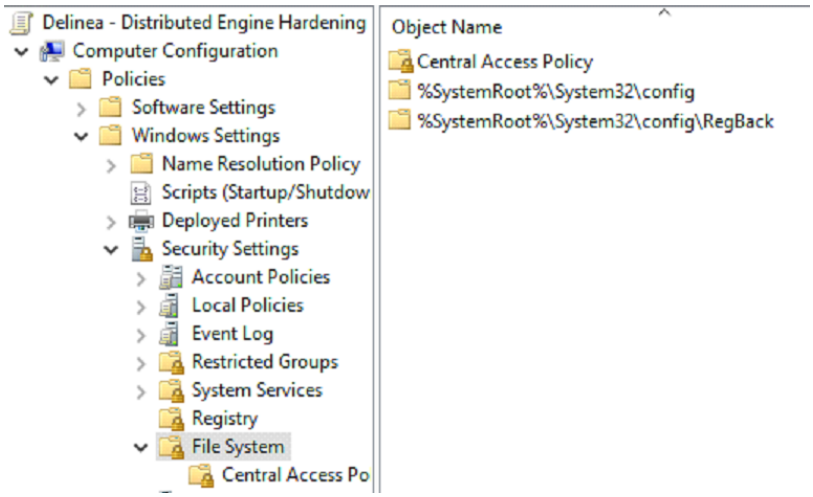
- Change Principal to Everyone, select Show Advanced Permissions, and select the following boxes:
- Traverse Folder\ Execute File
- List folder\ Read data
- Read attributes
- Read extended attribute



4. Click OK, then click Apply Settings.



5. Perform the same steps above for *SystemRoot%\System32\Config\RegBack*



Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies/Security Options

Policy	Recommended Value	Comment or Vulnerability
Accounts: Administrator account status	Enabled	
Accounts: Guest account status	Disabled	
Accounts: Limit local account use of blank passwords to console logon only	Enabled	
Accounts: Rename administrator account	It is recommended to change both the Administrator and the guest names to a name that will not testify about their permissions, and also to create a new locked and unprivileged user name Administrator as bate	Comment: Apply this parameter according to the organization security policy. Vulnerability: The administrators default name is known as a high privilege user. This user is a target for hacking attempts. Severity of the damage: Medium Operational aspects: None
Audit: Audit the use of Backup and Restore privilege.	Enabled	Vulnerability: The system does not monitor backup and restore activities of files, therefore it does not allow exposing unusual activities in this area. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Devices: Allowed to format and eject removable media	Administrator	Vulnerability: Users may be able to move data on removable disks to a different computer where they have administrative privileges. The user could then take ownership of any file, grant themselves full control, and view or modify any file. The fact that most removable storage devices will eject media by pressing a mechanical button diminishes the advantage of this policy setting. Severity of the damage: Low Operational aspects: None
Devices: Prevent users from installing printer drivers	Enabled	Vulnerability: A malicious user could install inappropriate printer drivers in a deliberate attempt to damage the computer, or a user might accidentally install malicious software that masquerades as a printer driver. Severity of the damage: Low Operational aspects: None
Domain member: Disable machine account password changes	Disabled	Vulnerability: Computers that cannot automatically change their account password are at risk from an attacker who could determine the password for the computer's domain account. Severity of the damage: Low Operational aspects: None
Domain member: Maximum machine account password age	30 days	Vulnerability: Setting this parameter to 0 will allow an attacker to execute Brute Force attacks to find the computer account password. Severity of the damage: Low Operational aspects: None
Domain member: Require strong (Windows 2000 or later) session key	Enabled	Vulnerability: Session keys that are used to establish secure channel communications between domain controllers and member computers are much stronger in Windows 2000 than they were in previous Windows operating systems. Severity of the damage: Low Operational aspects: None
Interactive logon: Do not display last user name	Enabled	Vulnerability: An attacker with access to the console (for example, someone with physical access or someone who is able to connect to the server through Remote Desktop Services, also known as Terminal Services) could view the name of the last user who logged on to the server. The attacker could then try to guess the password, use a dictionary, or use a brute force attack to try to log on. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Interactive logon: Do not require CTRL+ALT+DEL	Disabled	Vulnerability: If users are not required to press CTRL+ALT+DEL, they are susceptible to attacks that attempt to intercept their passwords, If this setting is enabled, an attacker could install a Trojan horse program that looks like the standard logon dialog box in the Windows operating system, and capture the user's password. Severity of the damage: Low Operational aspects: None
Interactive logon: Number of previous logons to cache (in case domain controller is not available).	0	Vulnerability: Users who access the server console will have their logon credentials cached on that server. An attacker who is able to access the file system of the server could locate this cached information and use a brute force attack to attempt to determine user passwords. Severity of the damage: Medium Operational aspects: The local Administrator password should be known in case of DC unavailability.

Policy	Recommended Value	Comment or Vulnerability
Interactive logon: Require Domain Controller authentication to unlock workstation	Enabled	Vulnerability: By default, the computer caches in memory the credentials of any users who are authenticated locally. The computer uses these cached credentials to authenticate anyone who attempts to unlock the console. When cached credentials are used, any changes that have recently been made to the account-such as user rights assignments, account lockout, or the account being disabled-are not considered or applied after the account is authenticated. User privileges are not updated, and (more important) disabled accounts are still able to unlock the console of the computer. Severity of the damage: Medium Operational aspects: The local Administrator password should be known in case of DC unavailability
Microsoft network client: Send unencrypted password to third-party SMB servers.	Disabled	Vulnerability: The server can transmit passwords in plaintext across the network to other computers that offer SMB services. These other computers might not use any of the SMB security mechanisms that are included with Windows Server 2003 and above. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Microsoft network server: Amount of idle time required before suspending session	15 minutes	Vulnerability: Each SMB session consumes server resources, and numerous null sessions will slow the server or possibly cause it to fail. An attacker could repeatedly establish SMB sessions until the server's SMB services become slow or unresponsive. Severity of the damage: Medium Operational aspects: None
Microsoft network server: Attempt S4U2Self to obtain claim information	Disabled	Vulnerability: Enabling this policy setting allows you take advantage of features in Windows Server 2012 and Windows 8 for specific scenarios to use claims-enabled tokens to access files or folders that have claim-based access control policy applied on Windows operating systems prior to Windows Server 2012 and Windows 8. Severity of the damage: Medium Operational aspects: None
Microsoft network server: Server SPN target name validation level	Off	Vulnerability: This policy setting controls the level of validation that a server with shared folders or printers performs on the service principal name (SPN) that is provided by the client computer when the client computer establishes a session by using the SMB protocol. The level of validation can help prevent a class of attacks against SMB servers (referred to as SMB relay attacks). This setting will affect both SMB1 and SMB2. Severity of the damage: Low Operational aspects: None
Network access: Do not allow anonymous enumeration of SAM accounts	Enabled	Vulnerability: An unauthorized user could anonymously list account names and shared resources and use the information to attempt to guess passwords or perform social engineering attacks. Severity of the damage: Medium Operational aspects: None
Network access: Do not allow anonymous enumeration of SAM accounts and shares	Enabled	Vulnerability: An unauthorized user could anonymously list account names and shared resources and use the information to attempt to guess passwords or perform social-engineering attacks. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Network access: Do not allow storage of passwords and credentials for network authentication.	Enabled	Vulnerability: Passwords that are cached can be accessed by the user when logged on to the computer. Although this information may sound obvious, a problem can arise if the user unknowingly runs malicious software that reads the passwords and forwards them to another, unauthorized user. Severity of the damage: Medium Operational aspects: This parameter could affect windows schedule task services
Network access: Let Everyone permissions apply to anonymous users	Disabled	Vulnerability: The system will allow all users, including users who have not identified themselves in the Domain, perform operations of reading information related to user accounts and the names of the shares. Severity of the damage: Medium Operational aspects: None
Network access: Named Pipes that can be accessed anonymously	List has been deleted	The policy was enabled and the existing list was deleted. Vulnerability: Ability to remotely access data on the system by an unauthorized user. Severity of the damage: Low Operational aspects: None
Network access: Remotely accessible registry paths	List has been deleted	The policy was enabled and the existing list was deleted. Vulnerability: An attacker could use information in the registry to facilitate unauthorized activities. To reduce the risk of such an attack, suitable ACLs are assigned throughout the registry to help protect it from access by unauthorized users. Severity of the damage: Low Operational aspects: None
Network access: Remotely accessible registry paths and subpaths	List has been deleted	The policy was enabled and the existing list was deleted. Vulnerability: An attacker could use information in the registry to facilitate unauthorized activities. To reduce the risk of such an attack, suitable ACLs are assigned throughout the registry to help protect it from access by unauthorized users. Severity of the damage: Low Operational aspects: None
Network access: Restrict anonymous access to Named Pipes and Shares.	Enabled	Vulnerability: Null sessions are a weakness that can be exploited through shared folders on computers environment. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Network access: Shares that can be accessed anonymously	List has been deleted	The policy was enabled and the existing list was deleted. Vulnerability: Any shared folders that are listed can be accessed by any network user, which could lead to the exposure or corruption of sensitive data. Severity of the damage: Medium Operational aspects: None
Network access: Sharing and security model for local accounts	Classic - Local users authenticate as themselves	Vulnerability: With the Guest only model, any user who can authenticate to the server over the network does so with guest privileges, which means that they will not have write access to shared resources on that server. Although this restriction does increase security, it makes it more difficult for authorized users to access shared resources on the server because ACLs on those resources must include access control entries (ACEs) for the Guest account. With the Classic model, local accounts should be password protected. Otherwise, if Guest access is enabled, anyone can use those user accounts to access shared system resources. Severity of the damage: Low Operational aspects: None
Network security: Do not store LAN Manager hash value on next password change	Enabled	Vulnerability: The SAM file can be targeted by attackers who seek access to user name and password hashes. Such attacks use special tools to discover passwords, which can then be used to impersonate users and gain access to resources on your network. Severity of the damage: Medium Operational aspects: None
Network security: Force logoff when logon hours expire	Enabled	Vulnerability: Users can remain connected to the computer outside of their allotted logon hours. Severity of the damage: Low Operational aspects: None
Network security: LAN Manager authentication level	Send NTLMv2 Responses Only/Refuse LM & NTLM	Vulnerability: The system allows identification of users in the old LM and NTLM protocols. The old identification protocols are vulnerable to attacks. Severity of the damage: Medium Operational aspects: These parameters could effect on legacy system if the system don't support NTLMv2

Policy	Recommended Value	Comment or Vulnerability
Network security: Minimum session security for NTLM SSP based (including secure RPC) clients	Require NTLMv2 session security Require 128-bit encryption	Vulnerability: Network traffic that uses the NTLM Security Support Provider (NTLM SSP) might be exposed such that an attacker who has gained access to the network can create man-in-the-middle attacks. Severity of the damage: Medium Operational aspects: None
Network security: Minimum session security for NTLM SSP based (including secure RPC) servers	Require NTLMv2 session security Require 128-bit encryption	Vulnerability: Network traffic that uses the NTLM Security Support Provider (NTLM SSP) might be exposed such that an attacker who has gained access to the network can create man-in-the-middle attacks. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Recovery console: Allow automatic administrative logon	Disabled	Vulnerability: The Recovery Console can be very useful when you need to troubleshoot and repair computers that do not start. However, it is dangerous to allow automatic logon to the console. Anyone could walk up to the server, disconnect the power to shut it down, restart it, select Recover Console from the Restart menu, and then assume full control of the server. Severity of the damage: Medium Operational aspects: None
Recovery console: Allow floppy copy and access to all drives and all folders	Disabled	Vulnerability: An attacker who can cause the system to restart into the Recovery Console could steal sensitive data and leave no audit or access trail. Severity of the damage: Low Operational aspects: None
Shutdown: Allow system to be shut down without having to log on	Disabled	Vulnerability: Users who can access the console locally could shut down the computer. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Shutdown: Clear virtual memory pagefile	Enabled	Vulnerability: Important information that is kept in real memory may be written periodically to the paging file to help the operating system handle multitasking functions. An attacker who has physical access to a server that has been shut down could view the contents of the paging file. The attacker could move the system volume into a different computer and then analyze the contents of the paging file. Although this process is time consuming, it could expose data that is cached from random access memory (RAM) to the paging file. Severity of the damage: Low Operational aspects: It takes longer to shut down and restart the computer, especially on computers with large paging files.
System Settings: Optional subsystems	No one	Enable the policy and delete the existing list of users that will be populated by default. Vulnerability: The POSIX subsystem introduces a security risk that relates to processes that can potentially persist across logons. If a user starts a process and then logs out, there is a potential that the next user who logs on to the computer could access the previous user's process. This would allow the second user to take actions on the process by using the privileges of the first user. Severity of the damage: Low Operational aspects: None
System settings: Use Certificate Rules on Windows Executables for Software Restriction Policies	Enable	Vulnerability: Without the use of software restriction policies, users and computers might be exposed to the running of unauthorized software, such as viruses and Trojans horses. Severity of the damage: Medium Operational aspects: None
User Account Control: Use Admin Approval Mode for the built-in Administrator account	Enable	Vulnerability: Malicious software running under elevated credentials without the user or administrator being aware of its activity. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
User Account Control: Allow UIAccess applications to prompt for elevation without using the secure desktop	Disable	Vulnerability: Without the use of software restriction policies, users and computers might be exposed to the running of unauthorized software, such as viruses and Trojans horses. Severity of the damage: Medium Operational aspects: None
User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode	Prompt for consent for non-Windows binaries	Vulnerability: Malicious programs running under elevated credentials without the user or administrator being aware of their activity. This setting raises awareness to the user that a program requires the use of elevated privilege operations and requires that the user be able to supply administrative credentials in order for the program to run. Severity of the damage: Medium Operational aspects: None
User Account Control: Behavior of the elevation prompt for standard users	Prompt for credentials on the secure desktop	Vulnerability: Malicious programs running under elevated credentials without the user or administrator being aware of their activity. This setting raises awareness to the user that a program requires the use of elevated privilege operations and requires that the user be able to supply administrative credentials in order for the program to run. Severity of the damage: Low Operational aspects: None
User Account Control: Run all administrator in admin approval mode	Enable	Vulnerability: This is the setting that turns on or off UAC. If this setting is disabled, UAC will not be used and any security benefits and risk mitigations that are dependent on UAC will not be present on the system. Severity of the damage: Medium Operational aspects: None
User Account Control: Switch to the secure desktop when prompting for elevation	Enable	Vulnerability: Elevation prompt dialog boxes can be spoofed, causing users to disclose their passwords to malicious software. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
User Account Control: Virtualize file and registry write failures to per-user locations	Enable	Severity of the damage: Low Operational aspects: None

Computer Configuration > Administrative Templates > Windows Components > Security Settings > Remote Desktop Services

Vulnerability:

An unlimited number of open connections can cause denial of Service attack on the Remote Desktop services, also known as Terminal Services.

If a disconnected session kept alive that can lead a session hijacking by an attacker.

Clipboard mapping enables the client to transfer a virus or a malicious application to the server as well as copy configuration or sensitive data from the server back to the client machine. There is a risk of infecting to the whole network or damaging the system.

Severity of the damage:

Medium

Operational aspects:

None

Path	Policy	Recommended Value
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Connections	Automatic reconnection	Disabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Connections	Configure keep-alive connection interval	Enabled Keep-Alive interval:1
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Connections	Deny logoff of an administrator logged in to the console session	Enabled

Path	Policy	Recommended Value
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection	Do not allow Clipboard redirection	Disabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection	Do not allow supported Plug and Play device redirection	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection	Do not allow COM port redirection	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection	Do not allow LPT port redirection	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection	Do not allow drive redirection	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Security	Do not allow local administrators to customize permissions	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Temporary Folders	Do not delete temp folders upon exit	Disabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Temporary Folders	Do not use temporary folders per session	Disabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Session Time Limits	End session when time limits are reached	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Remote Session Environment	Remove "Disconnect" option from Shut Down dialog	Enabled

Path	Policy	Recommended Value
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Remote Session Environment	Remove Windows Security item from Start menu	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Security	Require secure RPC communication	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Security	Set client connection encryption level	Enabled Encryption Level: High Level
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Connections	Set rules for remote control of Remote Desktop Services user sessions	Enabled View Session without user's permission
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Session Time Limits	Set time limit for active but idle Remote Desktop Services sessions	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Session Time Limits	Set time limit for disconnected sessions	Enabled 15 minutes

Computer Configuration > Policies > Windows Settings > Security Settings > User Rights Assignment

Policy	Recommended Value	Comment
Access Credential Manager as a trusted caller		Vulnerability: If an account is given this right, the user of the account can create an application that calls into Credential Manager and is then provided the credentials for another user. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment
Access this computer from the network	BUILTIN\Administrators	Vulnerability: This right allows the users to use the SMB communications protocol in front of the server. This protocol allows access to the operating resources, such as: sharing and remote system administration using the operating system's built-in tools. Severity of the damage: Medium Operational aspects: None
Act as part of the operating system		Vulnerability: Users with the Act as part of the operating system user right can take complete control of the computer and erase evidence of their activities. Severity of the damage: Medium Operational aspects: None
Adjust memory quotas for a process	NT AUTHORITY\NETWORK SERVICE, NT AUTHORITY\LOCAL SERVICE, BUILTIN\Administrators	Vulnerability: A user with the Adjust memory quotas for a process user right can reduce the amount of memory that is available to any process, which could cause business-critical network applications to become slow or to fail. This privilege could be used to start a denial-of-service (DoS) attack. Severity of the damage: Medium Operational aspects: None
Allow log on locally	BUILTIN\Administrators	Vulnerability: Any account with the Allow log on locally user right can log on at the console of the computer. If you do not restrict this user right to legitimate users who must log on to the console of the computer, unauthorized users could download and run malicious software to elevate their privileges. Severity of the damage: Medium Operational aspects: None
Allow log on through Remote Desktop Services	BUILTIN\Administrators	Vulnerability: Any account with the Allow log on through Remote Desktop Services user right can log on to the remote console of the computer. If you do not restrict this user right to legitimate users who must log on to the console of the computer, unauthorized users could download and run malicious software to elevate their privileges. Severity of the damage: Medium Operational aspects: None
Back up files and directories	BUILTIN\Administrators	Vulnerability: Users who can back up data from a computer could take the backup media to a non-domain computer on which they have administrative privileges and restore the data. They could take ownership of the files and view any unencrypted data that is contained within the backup set. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment
Bypass traverse checking	NT AUTHORITY\ NETWORK SERVICE, NT AUTHORITY\LOCAL SERVICE, NT AUTHORITY\Authenticated Users, BUILTIN\Administrators	Vulnerability: This right allows the user to access files and partitions although he is not authorized to view files and change them. Severity of the damage: Medium Operational aspects: None
Change the system time	BUILTIN\Administrators, NT AUTHORITY\LOCAL SERVICE	Vulnerability: Users who can change the time on a computer could cause several problems. For example, time stamps on event log entries could be made inaccurate, an attacker who changes a computer's time may cause that computer to be unable to obtain or grant Kerberos protocol tickets. Severity of the damage: Medium Operational aspects: None
Change the time zone	BUILTIN\Administrator	Vulnerability: Changing the time zone represents little vulnerability because the system time is not affected. This setting merely enables users to display their preferred time zone while being synchronized with domain controllers in different time zones. Severity of the damage: Low Operational aspects: None
Create a token object		Vulnerability: A user account that is given this user right has complete control over the system, and it can lead to the system being compromised. Severity of the damage: High operational aspects: None
Create global objects	NT AUTHORITY\SERVICE, NT AUTHORITY\NETWORK SERVICE, NT AUTHORITY\LOCAL SERVICE, BUILTIN\Administrators	Vulnerability: Users who can create global objects could affect processes that run under other users' sessions. This capability could lead to a variety of problems, such as application failure or data corruption. Severity of the damage: Medium Operational aspects: None
Create permanent shared objects		Vulnerability: Users who have the Create permanent shared objects user right could create new shared objects and expose sensitive data to the network. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment
Create symbolic links	Administrators	Vulnerability: Users who have the Create symbolic links user right could inadvertently or maliciously expose your system to symbolic link attacks. Symbolic link attacks can be used to change the permissions on a file, to corrupt data, to destroy data, or as a DoS attack. Severity of the damage: Low Operational aspects: None
Debug programs	BUILTIN\Administrator	Vulnerability: The Debug programs user right can be exploited to capture sensitive computer information from system memory or to access and modify kernel or application structures. Some attack tools exploit this user right to extract hashed passwords and other private security information or to insert rootkit code. Severity of the damage: Low Operational aspects: None
Deny access to this computer from the network	BUILTIN\Guests	Vulnerability: Users who can log on to the computer over the network can enumerate lists of account names, group names, and shared resources. Users with permission to access shared folders and files can connect over the network and possibly view or modify data. Severity of the damage: Medium Operational aspects: None
Deny log on as a batch job	BUILTIN\Guests	Vulnerability: Accounts that have the Deny log on as a batch job user right could be used to schedule jobs that could consume excessive computer resources and cause a DoS condition. Severity of the damage: Medium Operational aspects: None
Deny log on as a service	BUILTIN\Guests	Vulnerability: Accounts that can log on as a service could be used to configure and start new unauthorized services, such as a keylogger or other malicious software. Severity of the damage: Medium Operational aspects: None
Deny log on locally	BUILTIN\Guests	Vulnerability: Any account with the ability to log on locally could be used to log on at the console of the computer. If this user right is not restricted to legitimate users who must log on to the console of the computer, unauthorized users might download and run malicious software that elevates their privileges. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment
Deny log on through Remote Desktop Services	BUILTIN\Guests	Vulnerability: Any account with the right to log on through Remote Desktop Services could be used to log on to the remote console of the computer. If this user right is not restricted to legitimate users who need to log on to the console of the computer, malicious users might download and run software that elevates their privileges. Severity of the damage: Medium Operational aspects: None
Enable computer and user accounts to be trusted for delegation	BUILTIN\Administrators	Vulnerability: Misuse of the Enable computer and user accounts to be trusted for delegation user right could allow unauthorized users to impersonate other users on the network. An attacker could exploit this privilege to gain access to network resources and make it difficult to determine what has happened after a security incident. Severity of the damage: Medium Operational aspects: None
Force shutdown from a remote system	BUILTIN\Administrators	Vulnerability: Any user who can shut down a computer could cause a DoS condition to occur. Therefore, this user right should be tightly restricted. Severity of the damage: Low Operational aspects: None
Generate security audits	NT AUTHORITY\NETWORK SERVICE, NT AUTHORITY\LOCAL SERVICE	Vulnerability: Accounts that can write to the Security log could be used by an attacker to fill that log with meaningless events. If the computer is configured to overwrite events as needed, attackers could use this method to remove evidence of their unauthorized activities. If the computer is configured to shut down when it is unable to write to the Security log and it is not configured to automatically back up the log files, this method could be used to create a DoS condition. Severity of the damage: Low Operational aspects: None
Increase scheduling priority	BUILTIN\Administrators	Vulnerability: Increasing the working set size for a process decreases the amount of physical memory that is available to the rest of the system. Severity of the damage: Low Operational aspects: None
Load and unload device drivers	BUILTIN\Administrators	Vulnerability: Device drivers run as highly privileged code. A user who has the Load and unload device drivers user right could unintentionally install malicious software that masquerades as a device driver. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Lock pages in memory	BUILTIN\Administrators	Vulnerability: Users with the Lock pages in memory user right could assign physical memory to several processes, which could leave little or no RAM for other processes and result in a DoS condition. Severity of the damage: Low Operational aspects: None
Manage auditing and security log	BUILTIN\Administrators	Vulnerability: Anyone with the Manage auditing and security log user right can clear the Security log to erase important evidence of unauthorized activity. Severity of the damage: Medium Operational aspects: None
Modify an object label		Vulnerability: Anyone with the Modify an object label user right can change the integrity level of a file or process so that it becomes elevated or decreased to a point where it can be deleted by lower-level processes. Either of these states effectively circumvents the protection offered by Windows Integrity Controls and makes your system vulnerable to attacks by malicious software. Severity of the damage: Low Operational aspects: None
Modify firmware environment values	BUILTIN\Administrators	Vulnerability: Anyone who is assigned the Modify firmware environment values user right could configure the settings of a hardware component to cause it to fail, which could lead to data corruption or a DoS condition. Severity of the damage: Medium Operational aspects: None
Perform volume maintenance tasks	BUILTIN\Administrators	Vulnerability: A user who is assigned the Perform volume maintenance tasks user right could delete a volume, which could result in the loss of data or a DoS condition. Also, disk maintenance tasks can be used to modify data on the disk such as user rights assignments that might lead to escalation of privileges. Severity of the damage: Low Operational aspects: None
Profile single process	BUILTIN\Administrators	Vulnerability: The Profile single process user right presents a moderate vulnerability. Attackers with this user right could monitor a computer's performance to help identify critical processes that they might want to attack directly. Attackers may be able to determine what processes run on the computer so that they could identify countermeasures that they may need to avoid, such as antivirus software or an intrusion-detection system. They could also identify other users who are logged on to a computer. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Restore files and directories	BUILTIN\Administrators	Vulnerability: An attacker with the Restore files and directories user right could restore sensitive data to a computer and overwrite data that is more recent, which could lead to loss of important data, data corruption, or a denial-of-service condition. Attackers could overwrite executable files that are used by legitimate administrators or system services with versions that include malicious software to grant themselves elevated privileges, compromise data, or install programs that provide continued access to the computer. Severity of the damage: Medium Operational aspects: None
Shut down the system	BUILTIN\Administrators	Vulnerability: The ability to shut down the server should be limited to a very small number of trusted administrators. Severity of the damage: Low Operational aspects: None
Take ownership of files or other objects	BUILTIN\Administrators	Vulnerability: Any users with the Take ownership of files or other objects user right can take control of any object, regardless of the permissions on that object, and then make any changes that they want to make that object. Such changes could result in exposure of data, corruption of data, or a DoS condition. Severity of the damage: High Operational aspects: None

Distributed Engines Operations

Secret Server Operations

Secret Server is a message-based system where components can publish or consume messages. The transport of these messages is via a bus:

- Azure Service Bus for Secret Server Cloud
- RabbitMQ for Secret Server On-Premises

Message Processing

- Messages are published to a queue and consumers of those queues will remove the message and process the work associated with the message.
- Messages have internal (non-configurable) priorities, for instance, Directory Services authentication has a higher priority than HB.
- Messages can fall into one of three categories:
  1. One-way workflow: Single message, for instance, change DE logging levels.
  2. Two-way workflow asynchronous: First message performs HB, second message HB response.

3. Two-way workflow synchronous: First message DE requests a status update of a proxied session, SS engine worker sends a relatively immediate second message response.

Code Functionality

- The code limits the amount of messages it can consume in a single sitting before attempting to consume additional messages.
- No message routing to specific DEs is implemented.
- No equal distribution of various workloads to individual DEs is attempted.

Primary Architectural Goal

- High availability and allowing multiple endpoints to process workloads.

Distributed Engine Configuration

- Via site and engine settings, a DE can be configured to consume all message types or a subset of the message types.
- A DE can only consume messages from a single site.

Summary

- DE message consumption is a non-deterministic algorithm; the bus manages which consumer will consume a message.
- There is no round-robin distribution.
- There is no mechanism in place for the SSH/RDP proxy workflow to choose a specific DE for a business user.

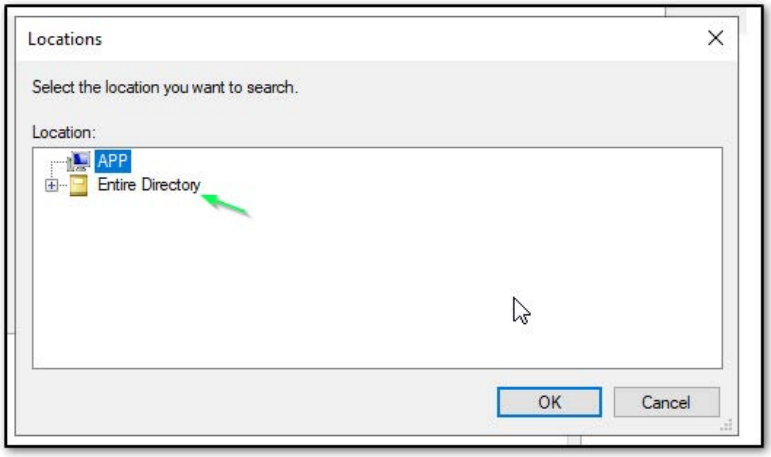
Distributed Engine Offline and Online Events

Delinea defines the definition of distributed engines' offline status to be the configured heartbeat interval times three. For instance, if your heartbeat interval is configured at 5 minutes, the engine will report offline if Secret Server and the engine do not successfully communicate within a 15-minute time period. Engine online and offline states were also added to subscription actions to allow notification to admins when engine states change.

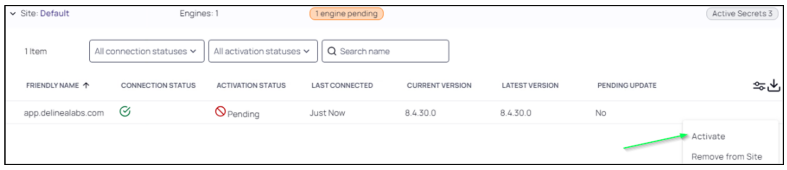
Downloading and Installing a Distributed Engine

1. Navigate to **Admin > Distributed Engine** and click **Add Engine**.
2. Select **Default** for the Preconfigured Site and click **Download Now**.
3. Open the Downloads directory on your machine and extract the *Thycotic.DistributedEngine.Service.Default.x64 zip*.
4. Open the unzipped folder and run setup.exe.

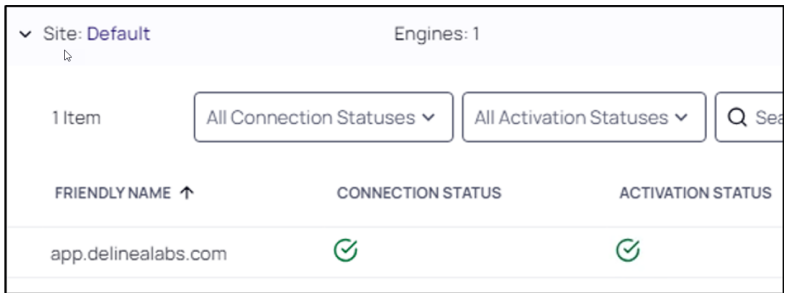
5. Open Services on the App Server and right-click the Thycotic Distributed Engine. Make sure that you use a service account with the least amount of privileges or level of access.
6. Select **Properties** and click the **Log On** tab.
7. Click **This account** and click **Browse**.
8. Click **Locations** and select **Entire Directory**.



9. Type the Service Account you would like to run in the Enter the object text box, check Names, then click **Ok**.
10. Switch back to your browser and reload the page.
11. Enter the Password for the Administrator and click **Ok**.
12. Switch back to your browser and reload the page.
13. Expand the **Default** site.
14. Hover over the Engine that is now showing, click the three dots menu, select **Activate** and click **OK** in the pop-up.



15. Green checkmarks should appear for Connection and Activation Status and your Engine has now been installed.



Facilitating Auto Upgrades of Your Distributed Engine

We have seen issues in the past getting engines auto-upgraded when the service is running as a domain account instead of the built-in account. Our Product Management team determined what is required to facilitate the auto upgrade for this scenario which is outlined below. From our research and testing, the important things that are needed to use an AD service account and to have successful, hands-free upgrades are the following:

1. Service account should be in the local Administrators group (so the account can start/stop the service).
2. Method 2 should be implemented in Group Policy editor from the Microsoft KB article
3. Service account needs full permissions to the DE installation directory
4. After setting the service account as the DE logon account, the service must be restarted to have the logon identity change take effect

After changing the service account, the DE will show up as a Pending engine. The old engine with the local logon identity should be deactivated and removed from the Site, and the new engine/identity instance added back to the Site upon activation.

Configuration and Sizing

> The distributed engine sizing guide is in development and will be available here when completed.

Requirements

Windows Server 2012

Starting in Secret Server version 8.9.000000, DEs require that one of following two server features be installed when the Secret Server website is running on a Windows Server 2012. This depends on which protocol is selected in the engine's callback settings. If HTTPS is selected, the HTTP activation is required. If TCP is selected, then TCP activation is required. This accomplished by going to one of the following in Windows Server 2012:

- **.NET Framework 4.5 Features > WCF Services > HTTP Activation**
- **.NET Framework 4.5 Features > WCF Services > TCP Activation**

If the feature is not installed, there will be an error message in the DE logs: `(405) Method Not Allowed. ---> System.Net.WebException: The remote server returned an error: (405) Method Not Allowed.`

Distributed Engine Installation

All interaction between the Secret Server Cloud tenant and your on premises network uses our distributed engine service to communicate. The work tasks that distributed engine completes includes Active Directory authentication, password changing, and heartbeat. The machine where the engine is installed must be able to communicate outbound on port 443.

For more information, see the "Distributed Engine Overview" on page 723.

To install the Distributed Engine:

1. Navigate to **Admin > Distributed Engine**
2. Click the **Add Engine** button, and in the Download Engine window select the related Processor Architecture for either 64-bit or 32-bit, and select the related Preconfigured Site. Click **Download now**.

   You can install distributed engine on your workstation or laptop for testing purposes, but for production installs, the distributed engine server should be installed on a server. Secret Server uses the distributed engine to communicate with your domain, so if your machine is turned off, users cannot log on with their domain accounts, and heartbeat and remote password changing will fail.

3. Run setup.exe as an administrator to install the engine service. This will install into `Thycotic Software Ltd\Distributed Engine`.
4. Go to **Admin > Distributed Engine**.
5. Under the **Sites and Engines** tab, expand the **Pending Engines** section. After you have installed an engine, it should appear here.
6. Select the engine by checking the box next to it, and select the related option - **Assign and Activate Selected Engines**, or **Assign Engines**. The Activate window will appear.
7. In the Site drop-down list select **New Site** to add a new site, **Default** - to add your default site, or select the related site from the list. Click **Activate**. The site with assigned engine will appear in the list of all the sites below. Expand it to view the details.
8. Validate the engine's connectivity:
   a. Under the Sites and Engines tab, click directly on the related site.
   b. On the Site page under the Site tab, click **Validate Connectivity**, then in the Validate Connectivity window set the related Timeout in seconds (how long in seconds to wait for a successful round trip from site to bus to engine back to site), and click **Validate**. It may take several minutes for the engine to register. If it does not immediately validate wait a few minutes and try again.

Configuring Engines for PowerShell Use

Secret Server can be configured to use PowerShell for several key task types:

- Discovery
- Secret Heartbeat

- Secret Password Change
- Script Tests

However, customers using PowerShell extensively for these tasks may experience high CPU usage, as PowerShell is more resource-intensive compared to cmd or bash. To address this, you can limit the number of "shells" (tasks) a user can run on a machine. These "shells" represent the tasks the engine is executing under the user account associated with the Engine Service.

This limit is controlled by the **MaxShellsPerUser** setting for WinRM, which PowerShell uses. You can configure this setting through Group Policy, the registry, or directly with PowerShell. If MaxShellsPerUser is set, the engine will only execute the allowed number of PowerShell tasks and queue any additional tasks in the Service Bus for later execution. If it's not set, the engine will run tasks as normal without restriction.

In cases where PowerShell scripts are unreliable or prone to timeouts—especially under heavy loads from Secret Server—the engine may encounter issues. Since the Distributed Engine allows scripts to run for up to 15 minutes, extended timeouts can cause problems. To mitigate this, you can use the Prefetch feature, which controls how many tasks the engine can consume at one time. For instance, you could limit the number of concurrent Secret Heartbeats by setting a specific prefetch value.

Additionally, a configuration setting allows you to match the prefetch count for all consumers that use PowerShell to the MaxShellsPerUser value. This setting can be found in the app-prefetch.config file located in the application directory.

General Networking

General networking topics, including site status, IP address filtering, clustering, and ports.

Checking Secret Server Site Status

To query Secret Server status without authentication for basic latency check, follow the steps below.

1. In a web browser, go to https://yoursecretserverurl/healthcheck.aspx
2. Compare the information displayed in your browser to the information below:

   ```
   {"healthy":true,"now":"2019-04-08T12:59:06.0455458-04:00","utcNow":"2019-04-
   08T16:59:06.0455458Z"}
   ```

   - If your information is similar, your Secret Server should be operational.
   - If your information displays other text such as **timed out** or **service unavailable**, there may be issues with the web site where the application is installed.

Restricting IP Addresses

IP address restrictions allow you to control which IP address ranges users can use to log in to Secret Server.

> This topic can be used with legacy Secret Server Cloud and Secret Server On-Premises, but does not apply to Secret Server Cloud with the Delinea Platform. When users log in to Secret Server Cloud, only the public IP address is provided to Secret Server.

> Only use Secret Server IP address if you are allowing direct logins to Secret Server Cloud.

If you overly locked down then you could block Delinea Platform access to Secret Server Cloud. Please reach out to Support for additional questions

Creating IP Address Ranges

To create an IP address range:

1. Go to **Admin > IP Addresses** under Administration. The IP Address Management page appears.
2. Click the **New IP Address** button. The Add New IP Address Range popup page appears.
3. In the **IP Address User/Network Name** text box, type a descriptive name for your range.
4. In the **IP Address Range** text box, enter an IP Address or IP Address range. Secret Server supports single IP Addresses (10.0.0.4), a range separated by a hyphen (10.0.0.1-10.0.0.255), and CIDR notation (10.0.0.0/24).
5. Click the **Save** button. The new address or range appears in the IP Address Management table.

You can show or hide columns in the table by clicking the  button.

Editing and Deleting IP Address Ranges

To edit an IP address range, go to the **IP Address Management** page, click on a range, and click **Edit**. To delete a range, click on the range and click the **Delete** button.

Assigning an IP Address Range

1. To assign a range to a user:
2. Go to **Admin > Users** page. The View User page appears.
3. Scroll to the bottom of the page and click the **Change IP Restrictions** button. The Edit IP Address Restrictions Page appears.
4. Click to select or deselect check boxes next to the ranges to choose which IP Addresses a user can use to access Secret Server. If no boxes are checked, the user can access Secret Server through any IP Address.
5. Click the **Save** button.

Regardless of the restrictions, users can always log in when accessing Secret Server on the server using a local IP address (127.0.0.1). This prevents total lockout from Secret Server.

To put in a whitelist restriction at the Web Application Firewall (WAF) layer, you need to put in a ticket to support to have that in place.

This topic applies to **Secret Server On-Premises** and standalone **Secret Server Cloud**.

Ports and IP Addresses Used by Secret Server

This article lists ports and addresses typically used in Secret Server.

Notes

Microsoft Remote Procedure Call (RPC) Functionality

- The Remote Procedure Call (RPC) dynamic port ranges are a range of ports used by RPC. This port range varies by operating system. For Windows Server 2008 or greater, this port range is 49152 to 65535 and this entire port range must be open for RPC technology to work. The RPC range is needed to perform Remote Password Changing (also RPC) since you will need to connect to the computer using DCOM protocol.
- The range can vary separately for MS Exchange servers. For more information about changing the Remote Procedure Call (RPC) port range, see How to configure RPC dynamic port allocation to work with firewalls.
- To see your ipv4 dynamic range on a given machine, type netsh int ipv4 show dynamicport tcp in the command line.
- To specify a specific port on your environment to communicate with, see the related article on enabling WMI ports on Windows client machines.

Source and Target Designations

- Secret Server On-Premises has an initial site named **local** that you can configure to perform actions within the environment using the application server or distributed engines.
- Configure local at **Settings > Configuration Search > Distributed Engine > Sites > <site name> > Processing Location** or **Settings > Configuration Search > Administration > Setup and Operation > Distributed Engine > Sites > <site name> > Processing Location.**
- Secret Server Cloud and Delinea Platform have an initial site named **Default** that performs all actions through distributed engines.

Port Listing

Active Directory Sync Ports

Table: Active Directory Sync Ports

Type of Traffic	Port Number	Source	Target	Purpose
Kerberos	TCP/88, UDP/88	Web server or engines assigned to site	Domain controllers	Authentication
LDAP	TCP/389, UDP/389	Web server or engines assigned to site	Domain controllers	AD sync, authentication, and authorization
LDAPS*	TCP/636, UDP/636	Web server or engines assigned to site	Domain controllers	AD sync, authentication, and authorization

Type of Traffic	Port Number	Source	Target	Purpose
SMB/Microsoft-DS	TCP445, UDP/445	Web server or engines assigned to site	Domain controllers	AD sync, authentication, and authorization

*For LDAPS to work, the LDAP port (389) must also be open.

Database Server Incoming Ports

Table: Database Server Incoming Ports

Type of Traffic	Port Number	Source	Target	Purpose
SQL connection	TCP/1433, UDP/1434	Web servers	Microsoft SQL Server or availbility group listener	Database communication. Can be customized

Discovery Ports

Table: Discovery Ports

Type of Traffic	Port Number	Source	Target	Purpose
RPC dynamic port range	TCP/49152-65535, UDP/49152-65535	Web server or engines assigned to site	Windows servers	Windows Server discovery scanning for user and service accounts
RPC endpoint mapper	TCP/135	Web server or engines assigned to site	Windows servers	Windows Server discovery scanning for user and service accounts
SMB/Microsoft-DS	TCP/445, UDP/445	Web server or engines assigned to site	Windows servers	Windows Server discovery scanning for user and service accounts
SSH	TCP/22	Web server or engines assigned to site	Unix and Linux Servers	Unix and Linux server discovery scanning for user and SSH keys

Distributed Engines and Application Servers

Table: Distributed Engines and Application Servers

Type of Traffic	Port Number	Source	Target	Purpose
RDP proxy	TCP/3390	Client workstations	Web server or engines assigned to site	Proxied RDP sessions
RDP Proxy Outbound	TCP/3389	Webserver or engines assigned to site	Windows servers	Proxied RDP sessions
SSH proxy	TCP/22	Client workstations	Web server or engines assigned to site	Proxied SSH sessions
SSH proxy	TCP/22	Webserver or engines assigned to site	Unix and Linux Servers	SSH proxy and terminal sessions
SSH terminal	TCP/22	Client workstations	Web server or engines assigned to site	SSH terminal traffic

Email Ports for On-Premise installations

Table: Email Ports for On-Premise installations

Type of Traffic	Port Number	Source	Target	Purpose
SMTP	TCP/25, TCP/465, TCP/2525, or TCP/587	Web servers	Mail servers	Email alerts and reporting. Port can be customized to suit environment

Message Queuing Site Connector Ports

Table: Message Queuing Site Connector Ports

Type of Traffic	Port Number	Source	Target	Purpose
MemoryMQ	TCP/8672 (non-SSL), TCP/8671 (SSL)	Web servers and distributed engines	MemoryMQ server	Non-production AMQP message queueing

Type of Traffic	Port Number	Source	Target	Purpose
RabbitMQ	TCP/5672 (non-SSL), TCP/5671 (SSL)	Web servers and distributed engines	RabbitMQ servers	AMQP message queueing
RabbitMQ CLI tools communication	TCP/25672-25682	Localhost	RabbitMQ nodes	Used by rabbitmqctl and other management tools to communicate with nodes.
RabbitMQ management interface	TCP/15672 or TCP/156771	Localhost	RabbitMQ nodes	HTTP API and management UI (not directly related to clustering). Can be configured to use HTTP or HTTPS

On-Premise Web Server Incoming Ports

Table: On-Premise Web Server Incoming Ports

Type of Traffic	Port Number	Source	Target	Purpose
HTTP	TCP/80	Client devices	Web server	Optional HTTP access for legacy devices to web application and API
HTTPS	TCP/443	Client devices	Web server	HTTPS access to web application and API
HTTPS	TCP/443	Distributed engine	Web server	Distributed engines use callback flow for initial activation, periodic check of site and site connector settings, log file uploads, and software updates. This callback can also be configured to communicate over HTTP or TCP

RabbitMQ Clustering Ports

Table: RabbitMQ Clustering Ports

Type of Traffic	Port Number	Source	Target	Purpose
EPMD (Erlang Port Mapper Daemon)	TCP/4369	All RabbitMQ cluster nodes	All RabbitMQ cluster nodes	Used for node discovery. The EPMD maps node names to network ports
Inter-node communication	TCP/25672-25682	All RabbitMQ cluster nodes	All RabbitMQ cluster nodes	Used for communication between cluster nodes, including data replication and heartbeats

RADIUS Server Ports

Table: RADIUS Server Ports

Type of Traffic	Port Number	Source	Target	Purpose
RADIUS authentication	UDP/1812	Web servers or Secret Server Cloud	Radius server	Authentication

Remote Password Changing Ports

Table: Remote Password Changing Ports

Type of Traffic	Port Number	Source	Target	Purpose
Encrypted Telnet	TCP/23, TCP/22, otTCP/992	Web server or engines assigned to site	iSeries Mainframes and zSeries Mainframes	RPC and heartbeat
Entra ID Microsoft Graph API	TCP/443	Web server or engines assigned to site	Entra Graph API	RPC and heartbeat
Kerberos password change	TCP/464, UDP/464	Web server or engines assigned to site	Microsoft Active Directory domain controllers	RPC and heartbeat

Type of Traffic	Port Number	Source	Target	Purpose
LDAP	TCP/389, UDP/389	Web server or engines assigned to site	Microsoft Active Directory domain controllers or LDAP-based domains	RPC and heartbeat
LDAPS	TCP/636, UDP/636	Web server or engines assigned to site	Microsoft Active Directory domain controllers or LDAP-based domains	RPC and heartbeat
Microsoft SQL	TCP/1433, UDP/1434	Web server or engines assigned to site	SQL servers	RPC and heartbeat
Oracle listener	TCP/1521	Web server or engines assigned to site	Oracle servers	RPC and heartbeat
RPC dynamic port range	TCP/49152-65535, UDP/49152-65535	Web server or engines assigned to site	Windows computers	RPC and heartbeat
RPC endpoint mapper	TCP/135	Web server or engines assigned to site	Windows servers	RPC and heartbeat
SMB/Microsoft-DS	TCP/445, UDP/445	Web server or engines assigned to site	Windows computers	RPC and heartbeat
SSH	TCP/22	Web server or engines assigned to site	SSH-based servers and devices	RPC and heartbeat

Type of Traffic	Port Number	Source	Target	Purpose
Sybase	TCP/2638, TCP/5000	Web server or engines assigned to site	Sybase Servers	RPC and heartbeat
Telnet	TCP/23	Web server or engines assigned to site	Networking equipment and Unix servers	RPC and heartbeat
Windows privileged account (WinNT ADSI service provider)	TCP/139	Web server or engines assigned to site	Windows servers	RPC and heartbeat

Secret Server Cloud Specific Traffic

Table: Secret Server Cloud Specific Traffic

Type of Traffic	Port Number	Source	Target	Purpose
Distributed engine communication	TCP/443 (Web Sockets) or TCP/443 (Web Sockets), TCP/5671, TCP/5672 (AMQP)	Distributed engine	Azure service bus	Message queueing

Syslog Ports

Table: Syslog Ports

Type of Traffic	Port Number	Source	Target	Purpose
Syslog	TCP/514, UDP/514	Web server or engines assigned to site	Syslog collector	SEIM logging

IP Addresses

Web Application Firewall (WAF) for Traffic to Secret Server Cloud

IP Address allow-listing is not necessary unless outbound firewall rules are in place. Generally, the public IP the hostname resolves to is based on geographical location of the request source. All IPs below should be allow-listed to ensure uninterrupted connectivity.

All regions:

- 45.60.32.37
- 45.60.34.37
- 45.60.36.37
- 45.60.38.37
- 45.60.40.37
- 45.60.104.37

Secret Server Cloud Outgoing IP Addresses

secretservercloud.com

Primary:

- 20.65.118.12
- 23.102.107.104
- 23.102.106.185
- 23.102.107.220
- 23.102.108.55
- 52.151.206.35
- 52.224.253.4
- 52.151.206.73
- 52.151.206.77
- 52.224.253.7
- 20.228.138.112/29

DR:

- 52.160.67.39
- 52.160.67.38
- 104.40.25.170
- 138.91.163.99
- 137.135.51.234
- 52.190.184.16/29

secretservercloud.co.uk

Primary:

- 20.0.46.111
- 51.142.243.172
- 20.0.46.112

- 20.0.46.123
- 20.0.46.124
- 20.162.162.64/29

Secondary:

- 51.104.62.220
- 51.104.62.213
- 51.104.63.38
- 51.104.62.185
- 51.104.62.252
- 20.117.16.40/29

secretservercloud.ca

Primary:

- 52.228.117.246
- 52.228.113.119
- 52.139.7.40
- 52.139.7.137
- 52.139.7.197
- 40.85.220.216/29

DR:

- 52.229.119.193
- 52.229.119.89
- 52.235.39.79
- 52.235.39.125
- 52.235.39.5
- 20.220.90.80/29

secretservercloud.eu

Primary:

- 20.79.64.213
- 20.79.65.3
- 20.79.226.78
- 20.79.226.180

- 20.79.226.116
- 51.116.178.152/29

DR:

- 20.50.180.242
- 20.50.180.187
- 20.50.154.28
- 20.50.176.86
- 20.50.156.219
- 20.16.113.88.144/29

secretservercloud.com.sg

Primary:

- 20.195.97.220
- 20.195.98.154
- 20.212.128.73
- 20.212.128.75
- 20.212.128.74
- 52.237.113.56/29

DR:

- 65.52.165.108
- 65.52.160.251
- 52.184.100.188
- 52.184.101.189
- 52.184.101.213
- 23.100.88.144/29

secretservercloud.com.au

Primary:

- 20.37.251.37
- 20.37.251.120
- 20.37.5.233
- 20.37.5.227
- 20.37.5.48
- 20.37.1.16/29

DR:

- 20.53.142.34
- 20.53.142.37
- 20.53.80.77
- 20.53.81.216
- 20.53.82.77
- 23.101.211.80/29

Azure Service Bus

To find the customer specific Azure service bus information navigate to `https://<tenantname>.secretservercloud.<tld>/AdminDiagnostics.aspx`.

Related Articles and Resources

How to configure RPC dynamic port allocation to work with firewalls (Microsoft)

Secret Server Clustering

This document is a guide to Delinea's Secret Server clusters for administrators and advanced users. Secret Server can run with multiple front-end Web servers. For a critical instance, clustering offers a redundant system to limit potential down time from a single point of failure. Clustering also allows users to load balance for better performance.

Overview

Clustering and Background Thread Changes in 10.7.

There are two major architectural changes in Secret Server 10.7:

> The first change is obvious in the Secret Server user interface, and the second is hidden but very important to those supporting Secret Server.

- **Primary Node:** We eliminated "primary nodes." Previously, some important background operations, such as password changing and heartbeat, would only run from the primary node. Now they run from all nodes. Given that, there is no longer a "Make Primary" button, and the ValidPrimaryNode setting no longer applies.
- **Background Operations:** There are no longer background threads for scheduled operations. Instead, operations are scheduled by Quartz.

Clustering Overview

With Secret Server clustering, you can easily scale Secret Server for redundancy and performance. Basic Secret Server clustering is simple–you install Secret Server and then copy the installation to another machine. Secret Server clustering has four core concepts or components:

Nodes

Each machine with Secret Server installed on it, pointing to the same database, is a *node*. All nodes respond to Web requests and thus are Web servers.

Backbone Bus

The backbone bus Internally handles all communication between the roles. In a clustered environment, the backbone bus should always be an installed RabbitMq messaging queue. This allows every node in the cluster to help with the workload. If the backbone bus is set to "internal," then each node is using its own internal backbone bus.

Engine Response Bus

The engine response bus facilitates communication from Secret Server to distributed engines and back.

Worker Roles

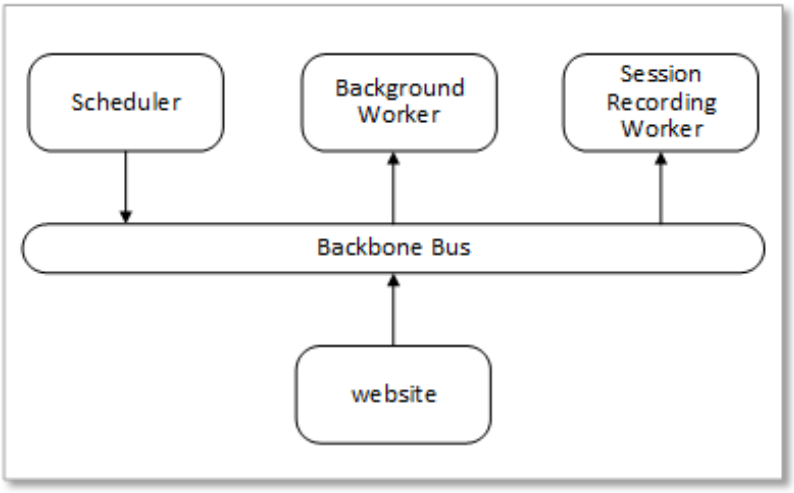
Each node can optionally run one or more worker roles: background Worker, engine worker, and session recording worker. Though they may run on the same machine, the roles do not directly communicate with each other.

Each node that is set to run the background worker role automatically runs the scheduler role as well. The scheduler role is responsible for running the vast majority of Secret Server background operations. It uses Quartz to run "trigger jobs" that send a message on the backbone bus for each scheduled operation. One or more background worker roles then processes those messages.

See the article "Troubleshooting Quartz Trigger Jobs" on page 260 for more information about Quartz.

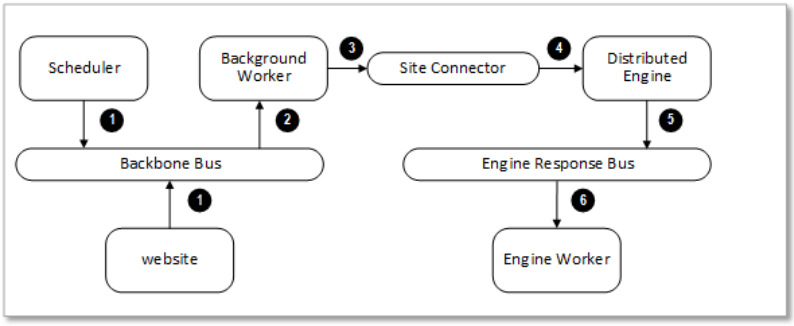
Component Communication

Figure: Secret Server Internal Cluster-Component Communication



Messages are placed on the backbone bus by the Scheduler role and the website. Messages are retrieved from the backbone bus.

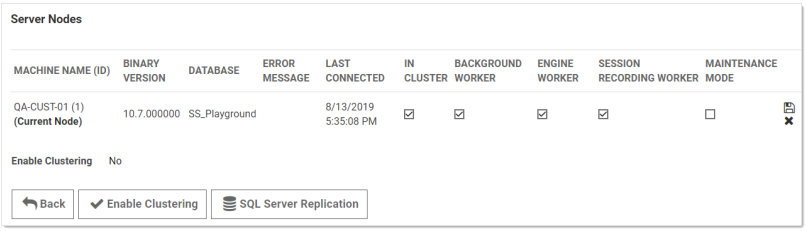
Figure: Secret Server Distributed Engine Communication



1. Manual or scheduled operation.
2. Background worker processes a message.
3. Outbound messages (password changes, heartbeats, and others) are placed on the site connector.
4. Distributed engine performs the operation.
5. Engine worker processes the response.

Server Node Configurations

The work an individual node handles depends entirely on which boxes are checked on the Server Nodes page (in edit mode):



- **In Cluster** is a toggle that turns a server node on or off. If enabled this node can process Web requests, and (if configured) will run the background, engine, and session recording roles. If disabled, the node is just a backup—it cannot run any roles, and trying to access the website on the node will redirect to the server nodes page.
- **Background Worker** is a toggle for all background operations, such as password changing, heartbeat, and discovery. When it is set to false, only the bulk operations, password generation, email, and secret import operations run on the node. See the list of background operations below.
- The **Background Worker**, **Engine Worker**, and **Session Recording Worker** check boxes enable the corresponding roles for that node.
- **Engine Worker** enables or disables the engine worker role, which processes responses from distributed engines.
- **Session Recording Worker** enables or disables the session recording role, which encodes session videos.
- **Maintenance Mode** enables or disables a read-only mode where the node cannot change secrets or related data.

Scheduled Background Operations

The current scheduled background operations in Secret Server are:

- ActiveDirectorySynchronizationMonitor
- BackgroundWorkerTaskTriggerJob
- BackupMonitor
- Bulk Operations When triggered by user
- CheckOutMonitor
- ComputerScanMonitor
- ConnectWiseMonitor
- DatabaseCleanupTriggerJob
- DiscoveryMonitor
- EventQueueMonitor
- ExpiredSecretPasswordChangeTriggerJob
- ExpiringLicenseTaskTriggerJob
- ExpiringSecretTaskTriggerJob
- HeartbeatMonitor
- Local Heartbeat Trigger Job
- Local Password Change Trigger Job
- NodeClusteringMonitor
- NodeTaskTriggerJob
- PasswordRequirementTriggerJob
- PbaDirectiveTriggerJob
- PbaMetadataUploadTriggerJob
- PrimaryNodeTaskMonitor
- Process Field Encryption Changes Task
- ProcessDashboardJsonValidationTask
- ProcessSecretPolicyChangesMessage
- ScheduledReportMonitor
- SecretComputerMatcherMonitor
- SecretItemHashMonitor
- SqlReplicationConflictMonitor
- TelemetryTriggerJob
- ThycoticOneSyncUserTriggerJob

- TruncateDatabaseCacheTriggerJob
- TruncateEngineLogTriggerJob
- VideoConversionTriggerJob

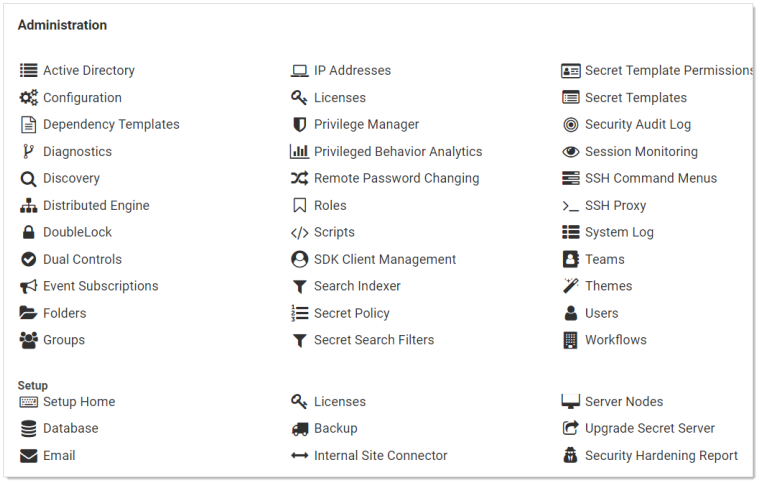
To see the current state of these jobs, such as the last time they ran and how long until they run again, go to **Admin > Diagnostics**.

Procedures

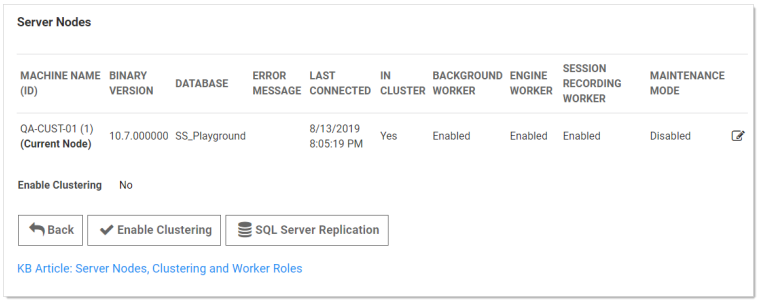
Setting up Clustering

> Clustering requires a Secret Server Premium add-on or Enterprise Plus edition license.

1. Have Secret Server upgraded or installed and running on a server.
2. Enable clustering on the node:
   a. In Secret Server, click **Admin > See All**. The Administration page appears:



   b. Click the **Server Nodes** button in the **Setup** section. The Server Nodes page appears:



   c. Click the **Enable Clustering** button.

3. Copy the entire Secret Server application folder (typically `c:\inetpub\wwwroot\SecretServer`) from the existing node to the secondary node.
4. Follow the steps in the Installation Guide for setting up the application pool and virtual directory in IIS.

   If you use DPAPI encryption for your encryption.config file, you need to transfer the non-DPAPI-encrypted version of the file to the secondary node. You can turn on DPAPI encryption from that server node locally after Secret Server is running. This setting can be found at **ADMIN > Configuration** on the **Security** tab.

5. If running Secret Server 8.9.300000 or later, ensure that both servers are using the same date and time.
6. Once the secondary server is running, navigate to its Secret Server using a Web browser.
7. Reset the database connection, following the instructions at "Changing SQL Server Connection Parameters" on page 175.
8. Activate licenses for the new node. You can do this on either server once the database connection is established on the secondary node.
9. Configure your load balancer for the two sites to have "sticky sessions" to prevent a user from bouncing between server on each request.
10. Configure the worker roles for the cluster:
    - Each server node can optionally run the background worker, engine worker, and session recording worker roles.
    - At least one instance of **each** type of those roles must be active in the cluster for the clustered Secret Server application to function.
    - You may run more than one instance of each role as desired to improve the performance of the clustered Secret Server application.

For more information on what the various roles do, please see the Worker Roles section.

Upgrading Secret Server in a Clustered Environment

Overview

Secret Server has a built-in Web installer. That installer is a series of pages inside Secret Server for downloading and updating Secret Server. Secret Server is accessible by users for most of the upgrade process. You can stop outside access to the site if you want to prevent users from making changes during the upgrade. Preventing user access will make restoring the database and site backups simpler if you decide to roll back the upgrade immediately afterward.

Before upgrading, **backup your Secret Server folder and database**. See Upgrading Secret Server - Single Instance and Web Clustering for important steps for ensuring your data is backed up.

Upgrading to Secret Server version 8.9.000000+ requires Windows Server 2008 R2 or greater.

> Upgrading to Secret Server 10.0.000000 and above requires configuring integrated pipeline mode on the Secret Server application pool. Please see "Changing IIS to Not Stop Worker Process in IIS 7.0 and Later" on page 238 for details.

> If using Integrated Windows authentication you will also need to update IIS authentication settings as detailed in "Configuring Integrated Windows Authentication" on page 383. If you are at version 9.1.000000 and below, you will need to first upgrade to 9.1.000001 before you can upgrade to 10.0.000000+.

> You do **not** need to download the Secret Server installer to perform an upgrade.

Procedure

1. Before you start:
   - Ensure that you have account credentials information and access for the server hosting Secret Server and the SQL Server instance hosting your Secret Server database.
   - Have a recent backup of the application files and database available.
   - Stop the application pools on all of the servers except the one that you have chosen to upgrade.
2. Choose one Secret Server server to upgrade
3. Perform a backup of that server.
4. Stop the Web servers of all other nodes.
5. Perform the upgrade using the same procedure as a single instance.

   > If applicable, see "Upgrading Secret Server Without Outbound Access" on page 149.

6. Once Secret Server is upgraded and working, copy the Web application folder (without the database.config or encryption.config files) to all other servers.

   > Never overwrite or delete the encryption.config file on a Secret Server server.

   > Both encryption.config and database.config are automatically propagated to the new servers from the original. If you need to copy those files because of database configuration changes and are using DPAPI, disable DPAPI encryption in Secret Server by going to **Admin > Configuration** on the **Security tab**. and clicking **Decrypt Key to not use DPAPI***before* copying those files to secondary servers. Please note, that DPAPI is system specific. Do not copy the database.config or the encryption.config from machine to machine if they are protected by DPAPI.

> EFS encryption is tied to the user account running the Secret Server application pool, so it is not machine-specific. Thus, it is not necessary to copy EFS encrypted files between Secret Server instances, but it is allowed.

7. If Delinea management server (TMS) is installed and clustered, copy the TMS directory to the secondary servers as well. The TMS directory is included by default for new installs of Secret Server 10.2+. TMS is used by advanced session recording and Privilege Manager. If the TMS folder and site does not exist in IIS, then no additional actions are needed.
8. Start the secondary servers to confirm they still work.

Upgrading Database Mirroring

1. If there is more than one Web server running Secret Server, ensure all instances are pointing to their primary database.
2. Select one server to perform the upgrade on, stop all other web servers.
3. Perform the upgrade on the single instance.
4. Once upgraded and working, copy the Web application folder to all other Web servers.
5. Start all other Web servers and confirm they work
6. Ensure all instances are properly activated
7. Ensure that the primary database changes have been replicated to the mirror database.
8. If one of the servers was pointing originally to the secondary database, adjust it to point there again.

Upgrading Disaster Recovery Installations

> This section does not refer to Resilient Secrets disaster recovery nodes.

1. Perform the upgrade on the production instance.
2. Backup the production instance.
3. Copy the database backup to the remote DR instance and restore the database.
4. Once the database is upgraded and working, copy the web application folder (but not the database.config or encryption.config files) to the remote DR instance, overwriting the existing files.
5. Restart IIS or recycle the application pool running Secret Server on the remote DR instance.
6. Confirm that the remote DR instance is working correctly.

Load Balancing Secret Server Clusters

In a clustered Secret Server environment set up behind a load balancer, the accessible outside URL may be something other than the server name.

> Customers often configure the service monitor on the load balancer to hit `login.aspx`. This can cause performance issues and memory leaks. We strongly suggest hitting `healthcheck.aspx` instead.

Custom URL Configuration

In Secret Server 8.5 and later, the Custom URL setting can be configured to ensure that links and resources are resolved correctly and are not based upon the server name. The Custom URL sets a definitive URL for Secret Server. Without it, features in Secret Server that need to build a link back to the server must construct the link using the host value on the request, which is susceptible to manipulation.

1. Navigate to **Admin > Configuration**.
2. On the **General** tab, click the **Edit** button.
3. Go to the **Application Settings** section.
4. Click to select the **Custom URL** check box.
5. Type the desired URL in the **Secret Server Custom URL** text box.

SSL Recommendations

For the best security, we recommend placing the SSL certificate on each of the Web servers. This ensures the traffic leaving the server is encrypted by SSL. Optionally, the load balancer would need the certificates as well for adding the client's IP address.

If the connection between the load balancer and the server is isolated in a security zone, you could leave HTTP between the load balancer and the server and have the SSL on the load balancer.

Configuring Client's IP Address (X-Forwarded-For)

Routing traffic through a load balancer will cause Secret Server to audit the IP address of the load balancer instead of the business user. To avoid this:

First, configure the load balancer to pass along the client's IP address in the header.

Then add the `appSettings` key `IpAddressHeader` with the value of the name of the header field containing the client's IP address. This setting must be added to **all** Secret Server Web servers. Depending on your Secret Server version, do this in one of two ways:

For Secret Server prior to 10.5.000000:

In the `web-appSetting.config` file in your Secret Server directory, add the following key:

```
<?xml version="1.0" encoding="utf-8" ?>
<appSettings>
  <add key="IpAddressHeader" value="X-Forwarded-For" />
</appSettings>
```

For Secret Server 10.5.000000 and later:

1. Go to `https://<SecretServerAddress>/ConfigurationAdvanced.aspx`.
2. Scroll to the bottom and click **Edit**.
3. Locate the **IP Address Header** text box, type `X-Forwarded-For`.
4. Click the **Save** button.

> The SSL certificate needs to exist on the load balancer and the Web server to ensure it has access to add the client IP address header.

Clustering Errors

The following error may arise when setting up or operating Secret Server clustering:

- Server dates do not match: If the Web server dates do not match, the audit records could be bad. The fix is to set the servers to the same time.

  > This only applies to Secret Server before version 8.9.300000.

- Secret Server version does not match: If some of the cluster nodes have been upgraded and others have not, their versions will conflict, producing this error. Nodes which have the wrong (older) version will not function correctly. To fix this issue, ensure that all the nodes in your cluster are upgraded. For nodes that are having this issue, you can copy the application folder (minus the database.config file) to replace the server files with the correct version.

HTTP

Topics related to HTTP in Secret Server.

Secret Server Support for HTTP/2

HTTP/2 is supported in IIS 10. HTTP/2 is handled within IIS, so this is primarily a Microsoft question in regards to compatibility. Please see HTTP/2 on IIS. At the end of this article, it clarifies when HTTP/2 is not supported

Secret Server does support Windows Integrated Authentication where a user's windows session is passed through for authentication to Secret Server. That is, there is no log on page for Secret Server. The majority of our customers are (and the default configuration for Secret Server is) using forms-based authentication with a log on page. Only the latter is HTTP/2 compliant.

HTTP/2 is only compatible with HTTPS protocol. Secret Server can also be configured to operate only on HTTPS (Admin > Configuration > Security > Force HTTPS/SSL), which we strongly recommend.

Securing Traffic with HTTP Strict Transport Security

HTTP Strict Transport Security (HSTS) is an additional security layer for HTTPS that ensures anybody accessing a given Web site or entity is forced to use HTTPS and not HTTP *prior* to making any HTTP requests, eliminating man-in-the-middle attacks. HSTS is an IETF Internet Standards Track protocol as specified in RFC 6797.

When the Force HTTPS/SSL option in enabled in Secret Server, the **Enable HSTS** check box is displayed. After the option is turned on, you can click the **Advanced** link to specify the **Maximum Age** in seconds, which is how long the policy is in affect before re-evaluating. The default value is 31536000 seconds or one year, which is also recommended value. We recommend that you set the value as high as possible, if the site should never be accessed without SSL. Even after HSTS is disabled, your browser automatically redirects to over SSL for the duration of the configured maximum age.

> We recommend using the IISReset command-line utility or restarting IIS in IIS manager after enabling the setting for the setting to take effect.

This feature is available in Secret Server version 8.6.000009 and higher and Password Reset Server version 4.0.000000 and higher.

> To see which browsers support HSTS, please refer to the Strict Transport Security page on the Can I Use website.

Messaging

> This topic only applies to **Secret Server On-Premises**.

Secret Server utilizes RabbitMQ as its primary messaging bus to facilitate efficient and secure communication between various components, such as distributed engines and site connectors. RabbitMQ ensures that operational instructions and data are passed asynchronously and securely, enhancing the system's scalability and reliability. Site connectors, which can be either RabbitMQ or MemoryMQ, act as intermediaries that hold work items for multiple sites and distribute these tasks among engines. This setup allows Secret Server to manage large-scale operations like password changes, heartbeats, and discovery processes efficiently. The use of RabbitMQ, with its robust message queuing capabilities, ensures that messages are encrypted during transit and provides high availability through clustering, making it a critical component for maintaining seamless and reliable operations in Secret Server environments.

Internal Site Connector

> This topic only applies to **Secret Server On-Premises**.

You can change how Secret Server processes messages by navigating to **Admin > See All** and selecting **Networking**.

Messages are generated and placed on the internal site connector, or backbone bus, every time a background operation is triggered whether by a schedule or on-demand.

The internal site connector receives and processes messages as a result of numerous actions:

- Bulk Operations
- Generate Password
- Secret Import (CSV and XML)
- Run Heartbeat Now
- Run Heartbeat (Scheduled)
- Run Password Change Now
- Run Password Change (Scheduled)
- Run Discovery Now

- Run Discovery (Scheduled)
- Run AD Sync Now
- Run AD Sync (Scheduled)
- Elements of Session Recording

The internal site connector, using the internal hosted bus, is adequate for bulk operations, heartbeat, discovery, and the like, but some Secret Server features, such as a clustered Web server node configuration or session recording, require a scalable messaging solution to boost processing performance. Our choice is RabbitMQ, which is an intermediary messaging broker that can handle large-scale message processing.

> For the highest scalability and reliability, Delinea recommends using RabbitMQ. MemoryMQ is an easier but less capable alternative and can be used for trials and proof of concepts but **should not** be used for production environments. Two exceptions are very small deployments and customers that do not use open-source software for compliance reasons.

The following is a typical internal hosted bus operation (for a bulk operation):

1. A Secret Server user triggers the a bulk operation.
2. A message is formed and sent over a TCP connection to the internal hosted bus.
3. Secret Server (on the same machine) receives the message.
4. Secret Server (on the same machine) processes the message.

We continually improve the internal hosted bus but still recommend RabbitMQ for a scalable performance boost. See "Installing RabbitMQ" on page 93 for more information.

RabbitMQ Durable Exchanges

> This topic only applies to **Secret Server On-Premises**.

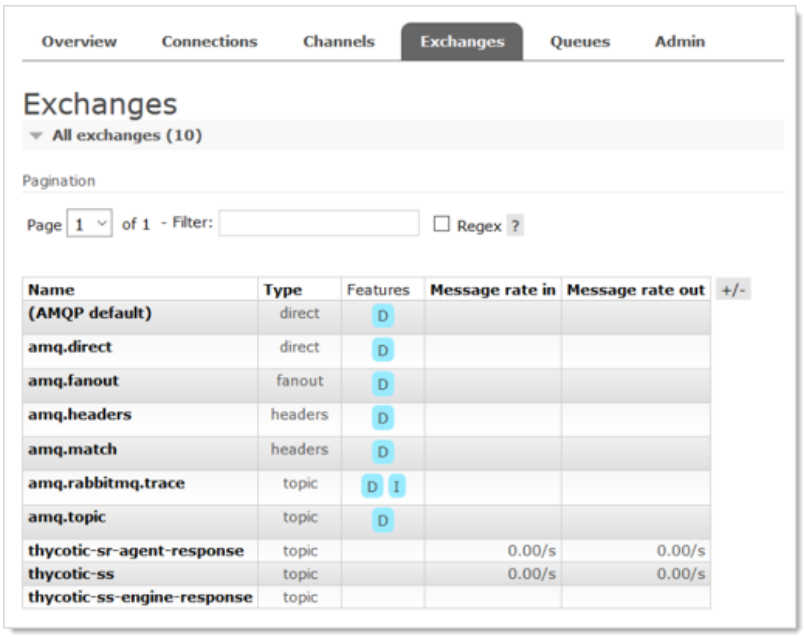
Overview

The Secret Server MessageQueue Client attempts to create RabbitMQ durable exchanges, logging the activity. A durable exchange is normally automatically re-created if RabbitMQ restarts for any reason. Any legacy non-durable exchanges disappear when RabbitMQ goes down and can only be manually recreated.

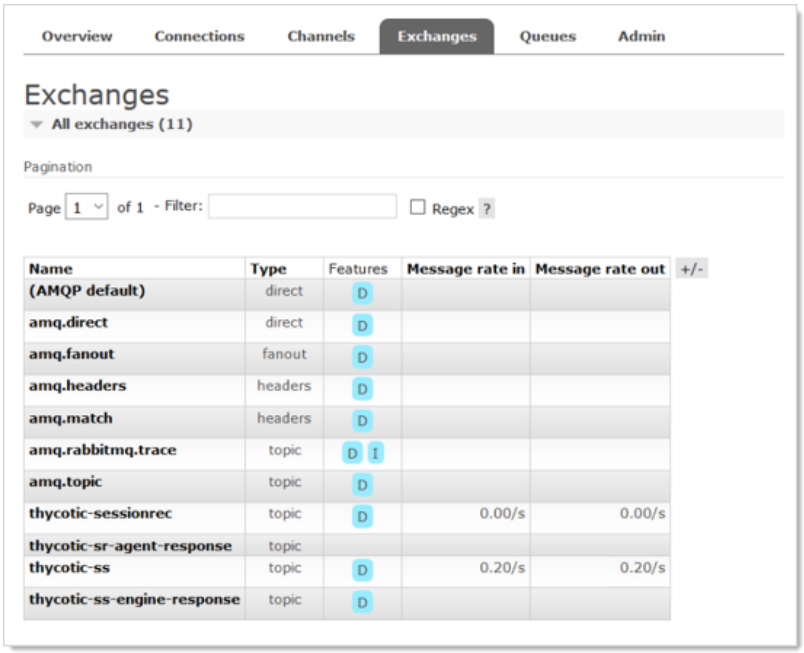
If the MessageQueue client detects that creating a durable exchange failed, it will log an error and attempt to create a non-durable one.

> Any existing non-durable exchanges, from previous versions of Secret Server, will also cause durable exchange creation to fail. See Manually Creating Durable RabbitMQ Exchanges.

Non-durable RabbitMQ exchanges for Secret Server would look similar to this, whether created by an earlier Secret Server version or by a durable-version-creation failure:



Note the absence of a 'D' in the Features column, meaning that exchange is not durable. Durable exchanges, created by the current Secret Server version (10.7.59+), look like this:



Earlier versions of Secret Server (before 10.7.59) created non-durable RabbitMQ exchanges during a Secret Server server or IIS restart. If the environment is clustered, the same is true of every node in that cluster. The current durable exchanges persist during any IIS restart, eliminating the need to restart Secret Server or recreate the exchanges.

However, any existing non-durable exchanges prevent the creation of the newer durable ones. To remedy that, you need to restart all of the RabbitMQ servers in the cluster at the same time or manually delete the non-durable exchanges.

Manually Creating Durable RabbitMQ Exchanges

To enjoy the benefits of the durable exchanges, you must first eliminate any legacy non-durable exchanges from your RabbitMQ server or servers. There are two ways to do this:

- Restart the RabbitMQ server or all of the RabbitMQ servers in the cluster at the same time. You can also stop the RabbitMQ service in `services.msc`.

Customers usually reset or turn off all servers via third party tools, but some prefer to shut off the service via services.msc because of their system configuration.

- Delete the exchanges manually:
  1. Click to select each Secret Server non-durable exchange, including distributed engine ones.
  2. Scroll to the bottom of the window.
  3. Click the **Delete** button.
  4. Restart all of the Secret Server instances and distributed engines to recreate the exchanges and connections.

Creating Durable RabbitMQ Exchanges with a PowerShell Script

Using the Script

```
powershell.exe -file exchangedurability.ps1 -username "guest" -password "guest" -
computerName "localhost" -port "15672"
```

The user has access to the RabbitMQ admin interface. The computername and port is where the admin interface is located.

The script:

1. Removes all of the exchanges that are not durable and any that are not the `thycotic-sr*` ones for legacy ASRAs.
2. Kills all of the connections. This forces the distributed engines and Secret Server to reconnect to the durable exchanges.

Script

```
param([string] $computerName = "",
    [string] $userName = "",
    [string] $password = "",
    [string] $port = ""
)
```

```
$defaultComputerName = if ($computerName -eq "") { "localhost" }else { $computerName }
$defaultVirtualhost = "/"
$defaultUserName = if ($userName -eq "") { "guest" }else { $userName }
$defaultPassword = if ($password -eq "") { "guest" }else { $password }
$defaultPort = if ($port -eq "") { "15672" }else { $port }
$defaultHttp = "http" #Use https if ssl
$defaultCredentials = New-Object System.Management.Automation.PSCredential
($defaultUserName, $(ConvertTo-SecureString $defaultPassword -AsPlainText -Force))
# LICENSE FOR LINKS - All the RabbitMQ PowerShell calls are based on this:
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/LICENSE
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/GetConnection.ps1
function Get-RabbitMQConnection
{
    [CmdletBinding(DefaultParameterSetName='defaultLogin', SupportsShouldProcess=$true,
ConfirmImpact='None')]
    Param
    (
        # Name of RabbitMQ Connection.
        [parameter(ValueFromPipeline=$true, ValueFromPipelineByPropertyName=$true)]
        [Alias("Connection", "ConnectionName")]
        [string[]]$Name = "",
        # Name of the computer hosting RabbitMQ server. Default value is localhost.
        [parameter(ValueFromPipelineByPropertyName=$true)]
        [Alias("HostName", "hn", "cn")]
        [string]$ComputerName = $defaultComputerName,
        # UserName to use when logging to RabbitMq server.
        [Parameter(Mandatory=$true, ParameterSetName='login')]
        [string]$UserName,
        # Password to use when logging to RabbitMq server.
        [Parameter(Mandatory=$true, ParameterSetName='login')]
        [string]$Password,
        # Credentials to use when logging to RabbitMQ server.
        [Parameter(Mandatory=$true, ParameterSetName='cred')]
        [PSCredential]$Credentials
    )
    Begin
    {
        $Credentials = NormaliseCredentials
    }
    Process
    {
        if ($pscmdlet.ShouldProcess("server $ComputerName", "Get connection(s):
$(NamesToString $Name '(all)')"))
        {
            $result = GetItemsFromRabbitMQApi -ComputerName
$ComputerName $Credentials "connections"
            $result = ApplyFilter $result 'name' $Name
            $result | Add-Member -NotePropertyName "ComputerName" -NotePropertyValue
$ComputerName
            SendItemsToOutput $result "RabbitMQ.Connection"
        }
    }
    End
    {
```

```
    }
}
#
https://github.com/mariuszwojcik/RabbitMQTools/blob/master/PreventUnEscapeDotsAndSlashesOn
Uri.ps1
if (-not $UnEscapeDotsAndSlashes) { Set-Variable -Scope Script -name
UnEscapeDotsAndSlashes -value 0x2000000 }
function GetUriParserFlags {
    $getSyntax = [System.UriParser].GetMethod("GetSyntax", 40)
    $flags = [System.UriParser].GetField("m_Flags", 36)
    $parser = $getSyntax.Invoke($null, "http")
    return $flags.GetValue($parser)
}
#
https://github.com/mariuszwojcik/RabbitMQTools/blob/master/PreventUnEscapeDotsAndSlashesOn
Uri.ps1
function SetUriParserFlags([int]$newValue) {
    $getSyntax = [System.UriParser].GetMethod("GetSyntax", 40)
    $flags = [System.UriParser].GetField("m_Flags", 36)
    $parser = $getSyntax.Invoke($null, "http")
    $flags.SetValue($parser, $newValue)
}
#
https://github.com/mariuszwojcik/RabbitMQTools/blob/master/PreventUnEscapeDotsAndSlashesOn
Uri.ps1
function PreventUnEscapeDotsAndSlashesOnUri {
    if (-not $uriUnEscapesDotsAndSlashes) { return }
    Write-Verbose "Switching off UnEscapesDotsAndSlashes flag on UriParser."
    $newValue = $defaultUriParserFlagsValue -bxor $UnEscapeDotsAndSlashes
    SetUriParserFlags $newValue
}
#
https://github.com/mariuszwojcik/RabbitMQTools/blob/master/PreventUnEscapeDotsAndSlashesOn
Uri.ps1
function RestoreUriParserFlags {
    if (-not $uriUnEscapesDotsAndSlashes) { return }
    Write-Verbose "Restoring UriParser flags - switching on UnEscapesDotsAndSlashes flag."
    try {
        SetUriParserFlags $defaultUriParserFlagsValue
    }
    catch [System.Exception] {
        Write-Error "Failed to restore UriParser flags. This may cause your scripts to
behave unexpectedly. You can find more at get-help about_UnEsapingDotsAndSlashes."
        throw
    }
}
if (-not $defaultUriParserFlagsValue) { Set-Variable -Scope Script -name
defaultUriParserFlagsValue -value (GetUriParserFlags) }
if (-not $uriUnEscapesDotsAndSlashes) { Set-Variable -Scope Script -name
uriUnEscapesDotsAndSlashes -value (($defaultUriParserFlagsValue -
band $UnEscapeDotsAndSlashes) -eq $UnEscapeDotsAndSlashes) }
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/Invoke_RestMethodProxy.ps1
function Invoke-RestMethod {
```

```
[CmdletBinding(HelpUri = 'http://go.microsoft.com/fwlink/?LinkID=217034')]
param(
    [Microsoft.PowerShell.Commands.WebRequestMethod]
    ${Method},
    [Parameter(Mandatory = $true, Position = 0)]
    [ValidateNotNullOrEmpty()]
    [uri]
    ${Uri},
    [Microsoft.PowerShell.Commands.WebRequestSession]
    ${WebSession},
    [Alias('SV')]
    [string]
    ${SessionVariable},
    [pscredential]
    ${Credential},
    [switch]
    ${UseDefaultCredentials},
    [ValidateNotNullOrEmpty()]
    [string]
    ${CertificateThumbprint},
    [ValidateNotNull()]
    [System.Security.Cryptography.X509Certificates.X509Certificate]
    ${Certificate},
    [string]
    ${UserAgent},
    [switch]
    ${DisableKeepAlive},
    [int]
    ${TimeoutSec},
    [System.Collections.IDictionary]
    ${Headers},
    [ValidateRange(0, 2147483647)]
    [int]
    ${MaximumRedirection},
    [uri]
    ${Proxy},
    [pscredential]
    ${ProxyCredential},
    [switch]
    ${ProxyUseDefaultCredentials},
    [Parameter(ValueFromPipeline = $true)]
    [System.Object]
    ${Body},
    [string]
    ${ContentType},
    [ValidateSet('chunked', 'compress', 'deflate', 'gzip', 'identity')]
    [string]
    ${TransferEncoding},
    [string]
    ${InFile},
    [string]
    ${OutFile},
    [switch]
```

```
        ${PassThru},
        [switch]
        ${AllowEscapedDotsAndSlashes})
    begin {
        try {
            $outBuffer = $null
            if ($PSBoundParameters.TryGetValue('OutBuffer', [ref]$outBuffer)) {
                $PSBoundParameters['OutBuffer'] = 1
            }
            $wrappedCmd = $ExecutionContext.InvokeCommand.GetCommand
('Microsoft.PowerShell.Utility\Invoke-RestMethod',
[System.Management.Automation.CommandTypes]::Cmdlet)
            # check whether need to disable UnEscapingDotsAndSlases on UriParser
            $requiresDisableUnEscapingDotsAndSlashes = ($AllowEscapedDotsAndSlashes -
and $Uri.OriginalString -match '%2f')
            # remove additional proxy parameter to prevent original function from failing
            if ($PSBoundParameters['AllowEscapedDotsAndSlashes']) { $null =
$PSBoundParameters.Remove('AllowEscapedDotsAndSlashes') }
            $scriptCmd = { & $wrappedCmd @PSBoundParameters }
            $steppablePipeline = $scriptCmd.GetSteppablePipeline
($myInvocation.CommandOrigin)
            $steppablePipeline.Begin($PSCmdlet)
        }
        catch {
            throw
        }
    }
    process {
        try {
            # Disable UnEscapingDotsAndSlashes on UriParser when necessary
            if ($requiresDisableUnEscapingDotsAndSlashes) {
                PreventUnEscapeDotsAndSlashesOnUri
            }
            $steppablePipeline.Process($_)
        }
        finally {
            # Restore UnEscapingDotsAndSlashes on UriParser when necessary
            if ($requiresDisableUnEscapingDotsAndSlashes) {
                RestoreUriParserFlags
            }
        }
    }
    end {
        try {
            $steppablePipeline.End()
        }
        catch {
            throw
        }
    }
}
<#
.ForwardHelpTargetName Invoke-RestMethod
.ForwardHelpCategory Cmdlet
```

```
#>
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/GetRabbitMQCredentials.ps1
function GetRabbitMQCredentials {
    Param
    (
        [parameter(Mandatory = $true)]
        [string]$userName,
        [parameter(Mandatory = $true)]
        [string]$password
    )
    $secpasswd = ConvertTo-SecureString $password -AsPlainText -Force
    return New-Object System.Management.Automation.PSCredential ($userName, $secpasswd)
}
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/NamesToString.ps1
function NamesToString {
    Param
    (
        [string[]]$name,
        [string]$altText = ""
    )
    if (-not $name) { return $altText }
    return $name -join ';'
}
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/ApplyFilter.ps1
function ApplyFilter {
    Param (
        [parameter()]
        [PSObject[]]$items,
        [parameter(Mandatory = $true)]
        [string]$prop,
        [string[]]$name
    )
    if (-not $name) { return $items }
    # apply property filter
    $filter = @()
    foreach ($n in $name) { $filter += '$_.' + $prop + '-like "' + $n + '"' }
    $sb = [scriptblock]::create($filter -join ' -or ')
    return $items | ? $sb
}
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/NormaliseCredentials.ps1
function NormaliseCredentials() {
    switch ($PsCmdlet.ParameterSetName) {
        "defaultLogin" { return GetRabbitMqCredentials $defaultUserName $defaultPassword
}
        "login" { return GetRabbitMqCredentials $UserName $Password }
        "cred" { return $Credentials }
    }
}
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/SendItemsToOutput.ps1
function SendItemsToOutput {
    Param
    (
        [parameter()]
```

```
        [PSObject[]]$items,
        [parameter(Mandatory = $true)]
        [string[]]$typeName
    )
    foreach ($i in $items) {
        $i.PSObject.TypeNames.Insert(0, $typeName)
        Write-Output $i
    }
}
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/GetItemsFromRabbitMQApi.ps1
function GetItemsFromRabbitMQApi {
    [CmdletBinding(DefaultParameterSetName = 'login')]
    Param
    (
        [parameter(Mandatory = $true, ParameterSetName = 'login', Position = 0)]
        [string]$cn,
        [parameter(Mandatory = $true, ParameterSetName = 'login', Position = 1)]
        [string]$userName,
        [parameter(Mandatory = $true, ParameterSetName = 'login', Position = 2)]
        [string]$password,
        [parameter(Mandatory = $true, ParameterSetName = 'login', Position = 3)]
        [string]$fn,
        [parameter(Mandatory = $true, ParameterSetName = 'cred', Position = 0)]
        [string]$computerName,
        [parameter(Mandatory = $true, ParameterSetName = 'cred', Position = 1)]
        [PSCredential]$cred,
        [parameter(Mandatory = $true, ParameterSetName = 'cred', Position = 2)]
        [string]$function
    )
    Add-Type -AssemblyName System.Web
    #Add-Type -AssemblyName System.Net
    if ($PsCmdlet.ParameterSetName -eq "login") {
        $computerName = $cn
        $cred = GetRabbitMqCredentials $userName $password
        $function = $fn
    }
    Write-Output $computerName
    $url = $defaultHttp + "://$([System.Web.HttpUtility]::UrlEncode
($computerName)):$defaultPort/api/$function"
    Write-Verbose "Invoking REST API: $url"
    return Invoke-RestMethod $url -Credential $cred -DisableKeepAlive -
AllowEscapedDotsAndSlashes
}
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/GetExchange.ps1
function Get-RabbitMQExchange {
    [CmdletBinding(DefaultParameterSetName = 'defaultLogin', SupportsShouldProcess =
$true, ConfirmImpact = 'None')]
    Param
    (
        # Name of RabbitMQ Exchange.
        [parameter(ValueFromPipeline = $true, ValueFromPipelineByPropertyName = $true)]
        [Alias("ex", "Exchange", "ExchangeName")]
        [string[]]$Name = "",
```

```
        # Name of RabbitMQ Virtual Host.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("vh")]
        [string]$VirtualHost = "",
        # Name of the computer hosting RabbitMQ server. Defalut value is localhost.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("HostName", "hn", "cn")]
        [string]$ComputerName = $defaultComputerName,
        # UserName to use when logging to RabbitMq server.
        [Parameter(Mandatory = $true, ParameterSetName = 'login')]
        [string]$UserName,
        # Password to use when logging to RabbitMq server.
        [Parameter(Mandatory = $true, ParameterSetName = 'login')]
        [string]$Password,
        # Credentials to use when logging to RabbitMQ server.
        [Parameter(Mandatory = $true, ParameterSetName = 'cred')]
        [PSCredential]$Credentials
    )
    Begin {
        $Credentials = NormaliseCredentials
    }
    Process {
        if ($pscmdlet.ShouldProcess("server $ComputerName", "Get exchange(s):
$(NamesToString $Name '(all)')")) {
            $exchanges = GetItemsFromRabbitMQApi -ComputerName
$ComputerName $Credentials "exchanges"
            $result = ApplyFilter $exchanges 'vhost' $VirtualHost
            $result = ApplyFilter $result 'name' $Name
            $result | Add-Member -NotePropertyName "ComputerName" -NotePropertyValue
$ComputerName
            SendItemsToOutput $result "RabbitMQ.Exchange"
        }
    }
    End {
    }
}
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/RemoveExchange.ps1
function Remove-RabbitMQExchange {
    [CmdletBinding(DefaultParameterSetName = 'defaultLogin', SupportsShouldProcess =
$true, ConfirmImpact = "High")]
    Param
    (
        # Name of RabbitMQ Exchange.
        [parameter(Mandatory = $true, ValueFromPipeline = $true,
ValueFromPipelineByPropertyName = $true, Position = 0)]
        [Alias("Exchange", "ExchangeName")]
        [string[]]$Name,
        # Name of RabbitMQ Virtual Host.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("vh", "vhost")]
        [string]$VirtualHost = $defaultVirtualhost,
        # Name of the computer hosting RabbitMQ server. Defalut value is localhost.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("HostName", "hn", "cn")]
```

```
        [string]$ComputerName = $defaultComputerName,
        # UserName to use when logging to RabbitMq server.
        [Parameter(Mandatory = $true, ParameterSetName = 'login')]
        [string]$UserName,
        # Password to use when logging to RabbitMq server.
        [Parameter(Mandatory = $true, ParameterSetName = 'login')]
        [string]$Password,
        # Credentials to use when logging to RabbitMQ server.
        [Parameter(Mandatory = $true, ParameterSetName = 'cred')]
        [PSCredential]$Credentials
    )
    Begin {
        $Credentials = NormaliseCredentials
        $cnt = 0
    }
    Process {
        if ($pscmdlet.ShouldProcess("server: $ComputerName, vhost: $VirtualHost", "Remove
exchange(s): $(NamesToString $Name '(all)')")) {
            foreach ($n in $Name) {
                $url = $defaultHttp + "://$([System.Web.HttpUtility]::UrlEncode
($ComputerName)):$defaultPort/api/exchanges/$([System.Web.HttpUtility]::UrlEncode
($VirtualHost))/$([System.Web.HttpUtility]::UrlEncode($n))"
                Write-Output $url
                $result = Invoke-RestMethod $url -Credential $Credentials -
AllowEscapedDotsAndSlashes -DisableKeepAlive -ErrorAction Continue -Method Delete
                Write-Verbose "Deleted Exchange $n on server $ComputerName, Virtual Host
$VirtualHost"
                $cnt++
            }
        }
    }
    End {
        if ($cnt -gt 1) { Write-Verbose "Deleted $cnt Exchange(s)." }
    }
}
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/AddExchange.ps1
function Add-RabbitMQExchange {
    [CmdletBinding(DefaultParameterSetName = 'defaultLogin', SupportsShouldProcess =
$true, ConfirmImpact = "Medium")]
    Param
    (
        # Name of RabbitMQ Exchange.
        [parameter(Mandatory = $true, ValueFromPipeline = $true,
ValueFromPipelineByPropertyName = $true, Position = 0)]
        [Alias("Exchange", "ExchangeName")]
        [string[]]$Name,
        # Type of the Exchange to create.
        [parameter(Mandatory = $true, ValueFromPipelineByPropertyName = $true)]
        [ValidateSet("topic", "fanout", "direct", "headers")]
        [string]$Type,
        # Determines whether the exchange should be Durable.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [switch]$Durable,
```

```
        # Determines whether the exchange will be deleted once all queues have finished
using it.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [switch]$AutoDelete,
        # Determines whether the exchange should be Internal.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [switch]$Internal,
        # Allows to set alternate exchange to which all messages which cannot be routed
will be send.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("alt")]
        [string]$AlternateExchange,
        # Name of RabbitMQ Virtual Host.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("vh", "vhost")]
        [string]$VirtualHost = $defaultVirtualhost,
        # Name of the computer hosting RabbitMQ server. Defalut value is localhost.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("HostName", "hn", "cn")]
        [string]$ComputerName = $defaultComputerName,
        # UserName to use when logging to RabbitMq server.
        [Parameter(Mandatory = $true, ParameterSetName = 'login')]
        [string]$UserName,
        # Password to use when logging to RabbitMq server.
        [Parameter(Mandatory = $true, ParameterSetName = 'login')]
        [string]$Password,
        # Credentials to use when logging to RabbitMQ server.
        [Parameter(Mandatory = $true, ParameterSetName = 'cred')]
        [PSCredential]$Credentials
    )
    Begin {
        $Credentials = NormaliseCredentials
    }
    Process {
        if ($pscmdlet.ShouldProcess("server: $ComputerName, vhost: $VirtualHost", "Add
exchange(s): $(NamesToString $Name '(all)')")) {
            $body = @{
                type = "$Type"
            }
            if ($Durable) { $body.Add("durable", $true) }
            if ($AutoDelete) { $body.Add("auto_delete", $true) }
            if ($Internal) { $body.Add("internal", $true) }
            if ($AlternateExchange) { $body.Add("arguments", @{ "alternate-exchange" =
$AlternateExchange }) }
            $bodyJson = $body | ConvertTo-Json
            foreach ($n in $Name) {
                $url = $defaultHttp+"://$([System.Web.HttpUtility]::UrlEncode
($ComputerName)):$defaultPort/api/exchanges/$([System.Web.HttpUtility]::UrlEncode
($VirtualHost))/$([System.Web.HttpUtility]::UrlEncode($n))"
                Write-Verbose "Invoking REST API: $url"
                $result = Invoke-RestMethod $url -Credential $Credentials -
AllowEscapedDotsAndSlashes -DisableKeepAlive -ErrorAction Continue -Method Put -
ContentType "application/json" -Body $bodyJson
```

```
                Write-Verbose "Created Exchange $n on server $ComputerName, Virtual Host
$VirtualHost"
                $cnt++
            }
        }
    }
    End {
        if ($cnt -gt 1) { Write-Verbose "Created $cnt Exchange(s)." }
    }
}
# Modified to allow + in url.
# https://github.com/mariuszwojcik/RabbitMQTools/blob/master/RemoveQueue.ps1
function Remove-RabbitMQConnection {
    Param
    (
        # Name of RabbitMQ connection.
        [parameter(Mandatory = $true, ValueFromPipeline = $true,
ValueFromPipelineByPropertyName = $true, Position = 0)]
        [Alias("ConnectionName")]
        [string] $Name = "",
        # Name of the computer hosting RabbitMQ server. Defalut value is localhost.
        [parameter(ValueFromPipelineByPropertyName = $true)]
        [Alias("HostName", "hn", "cn")]
        [string]$ComputerName = $defaultComputerName,
        # Credentials to use when logging to RabbitMQ server.
        [Parameter(Mandatory = $false)]
        [PSCredential]$Credentials = $defaultCredentials
    )
    $url = $defaultHttp + "://$([System.Web.HttpUtility]::UrlEncode
($ComputerName)):$defaultPort/api/connections/$([System.Web.HttpUtility]::UrlEncode
($name))"
    $url = $url.Replace("+", "%20")
    Write-Output $url
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("X-Reason", "Removing To Create Durable Exchanges")
    $result = Invoke-RestMethod $url -Credential $Credentials -Headers $headers -
DisableKeepAlive:$InvokeRestMethodKeepAlive -ErrorAction Continue -Method Delete
    Write-Output "$url closed."
    Write-Verbose "Closed connection $n to server $ComputerName"
}
function MakeExistingExchangesDurable() {
    Param(
        [string] $HostName = $defaultComputerName,
        [string] $UserName = $defaultUserName,
        [string] $Password = $defaultPassword,
        [string] $VirtualHost = "/",
        [bool] $IgnoreConfirms = $false
    )
    $exchanges = Get-RabbitMQExchange
    $nondurableExchanges = New-Object System.Collections.ArrayList
    Foreach ($exchange in $exchanges) {
        if ($exchange.name -and -not ($exchange.durable) -and -
not $exchange.name.Contains("Delinea-sr")) {
```

```
            $nondurableExchanges.Add($exchange) > $null
        }
    }
    if ($nondurableExchanges.Count -eq 0) {
        Write-Output "All the exchanges are durable."
        return
    }
    Write-Output "rnFound these exchanges as not durable:"
    Write-Output $nondurableExchanges | ForEach-Object { '{0}' -f $_.Name }
    $confirmation = ''
    if ($IgnoreConfirms -eq $false) {
        $confirmation = Read-Host "Are you Sure You Want To Proceed [y/n]"
    }
    if ($confirmation -eq 'y' -or $IgnoreConfirms -eq $true) {
        try {
            Foreach ($nondurableExchange in $nondurableExchanges) {
                Remove-RabbitMQExchange -Name $nondurableExchange.Name -VirtualHost
$nondurableExchange.vhost -Confirm:$(-not $IgnoreConfirms)
                Add-RabbitMQExchange -Name $nondurableExchange.Name -Durable:$true -Type
$nondurableExchange.type -AutoDelete:$nondurableExchange.auto_delete -
Internal:$nondurableExchange.Internal -VirtualHost $nondurableExchange.vhost -Confirm:$(-
not $IgnoreConfirms)
            }
            $connections = Get-RabbitMQConnection
            Foreach ($connection in $connections) {
                if ($connection.Name)
                {
                    Remove-RabbitMQConnection $connection.Name
                }
            }
        }
        catch {
            throw $_
        }
        Write-Output "Exchanges are now durable."
    }
    else {
        Write-Output "Not going to make the exchanges durable."
    }
}
MakeExistingExchangesDurable -IgnoreConfirms $true
```

RabbitMQ Naming Conventions for Queues

This topic only applies to **Secret Server On-Premises**.

Introduction

This document addresses RabbitMQ naming conventions for its queues. These queues are useful for application troubleshooting and proactive application monitoring.

Secret Server is an asynchronous message-based system where operational instructions and data are passed back and forth between various components running in Web nodes or distributed Engines. A GUI interaction to perform an action, such as heartbeat, remote password change, or bulk operations publishes a message and then returns control back to the user. RabbitMQ is the message bus or broker that facilitates the message traffic.

All Secret Server messages are encrypted on the bus, so you cannot peek into the message contents during transit.

Messages have a lifetime, and consumers discard expired messages. Therefore, any accumulation or backup of messages in any queue is abnormal and indicative of an application problem.

Secret Server Roles

Secret Server divides its functionality by named and unnamed roles, and only named roles are configurable in a Web node via the GUI.

Table: Secret Server Roles Related to Message Queues

Role	Type	Comment
Background Worker	Named	Background work initiated by a task, schedule or UI interaction. Final action of the work might be done in the current Web node, another Web node or sent to a distributed Engine to complete.
Engine	Unnamed	Processes work related to but not limited to: Active Directory synchronization, discovery, heartbeat, and remote password change.
Engine Worker	Named	Processes the response sent back from an engine.
Session Recording Worker	Named	Background work dedicated to session recording processing.
UI	Unnamed	IIS/ASP.NET processing, inbound access controlled by a load balancer.
API	Unnamed	IIS/ASP.NET processing, inbound access controlled by a load balancer.

Queue Names

A queue name is divided into three major sections with a colon (:) delimiter between each section:

```
Section1:Section2:Section3
```

Section1

Section1 represents a RabbitMQ exchange name. There are three predetermined exchange names, two legacy predetermined exchange names, and then a variable number of exchanges determined by the number of Secret Server sites.

Table: RabbitMQ Exchange Names

Role	Exchange Name	Type	Consumer Component Location	Comment
Background Worker	thycotic-ss	Predetermined	Web Node	
Engine	Site Name	Variable	Web Node or Distributed Engine	The out-of-the-box local site can be configured for either a Web node or a distributed engine. Any other site name is processed by a distributed engine.
Engine Worker	thycotic-ss-engine-response	Predetermined	Web Node	
Session Recording Worker	thycotic-sessionrec	Predetermined	Web Node	
Session Recording Worker	thycotic-sr	Predetermined-Legacy	Web Node	Legacy: background work dedicated to session recording processing.
Session Recording Worker	thycotic-sr-agent-response	Predetermined-Legacy	Web Node	Legacy: processes data sent by an advanced session recoding agent.

Variable site exchanges: If the Secret Server site is called local, then `local:` will also be the exchange name. If the Site is called Mars, then `Mars:` will be the exchange name.

Section2

Section2 typically has a name which represents a functional area in Secret Server code that is a consumer of the message.

Section3

Section3 represents the message name.

Secret Server Roles and Queues

This section of the message associates roles with queues and breaks the down by product functionality. Functionality can span multiple roles, for example, discovery is done by background worker, engine and engine worker roles while event pipelines is only done by a background worker role.

Classic Queue Name

```
thycotic-ss-engine-
response:Thycotic.MessageQueue.Common.Consumers.AutomaticSink.CreateAutomaticSinkConsumer:Th
ycotic.MessageQueue.Common.Messages.AutomaticSink.Request.CreateAutomaticSinkMessage
```

Quorum Queue Name

```
QQ-thycotic-ss-engine-
response:Thycotic.MessageQueue.Common.Consumers.AutomaticSink.CreateAutomaticSinkConsumer:Th
ycotic.MessageQueue.Common.Messages.AutomaticSink.Request.CreateAutomaticSinkMessage
```

Background Worker Role Queues

List of queues for background worker's functional areas:

Active Directory Synchronization

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.ActiveDirectorySynchronization.SynchronizationConsumer:Thycotic.ihawu.Business.Messages.Areas.ActiveDirectorySynchronization.Request.RunNowSynchronizationMessage-SecretServer:
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ActiveDirectorySynchronization.SynchronizationConsumer:Thycotic.ihawu.Business.Messages.Areas.ActiveDirectorySynchronization.Request.SynchronizationMessage

Bulk Operation

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.BulkOperation.BulkOperationConsumer:Thycotic.ihawu.Business.Messages.Areas.BulkOperation.Request.BulkOperationMessage

ConnectWise Integration

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.ConnectWise.ConnectWiseConsumer:Thycotic.ihawu.Business.Messages.Areas.ConnectWise.Request.ConnectWiseMessage-SecretServer:
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ConnectWise.ConnectWiseConsumer:Thycotic.ihawu.Business.Messages.Areas.ConnectWise.Request.RunNowConnectWiseMessage

Discovery

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.Discovery.ComputerScanConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.ComputerScanMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.ComputerScanConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.RunNowComputerScanMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.DiscoveryConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.DiscoveryMessage

- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.DiscoveryConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.RunNowDiscoveryMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.DiscoveryRuleApplierConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.RunDiscoveryRuleApplierMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.SecretComputerMatcherConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.RunNowSecretComputerMatcherMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Discovery.SecretComputerMatcherConsumer:Thycotic.ihawu.Business.Messages.Areas.Discovery.Request.SecretComputerMatcherMessage

Duo Integration

- thycotic-SecretServer:Thycotic.ihawu.BackgroundWorker.Logic.Areas.Duo.DuoAuthConsumer:Thycotic.Messages.ihawu.Areas.Duo.DuoRequestMessage

Email Processing

- thycotic-SecretServer:Thycotic.ihawu.BackgroundWorker.Logic.Areas.Email.SendEmailConsumer:Thycotic.Messages.ihawu.Areas.Email.Request.SystemSendEmailMessage
- thycotic-ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.Email.VerifySendEmailConsumer:Thycotic.Messages.ihawu.Areas.Email.Request.VerifySendEmailRequest

Event Pipelines

- thycotic-SecretServer:Thycotic.ihawu.BackgroundWorker.Logic.Areas.EventPipelines.EventPipelineActivityConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.EventPipelineActivityEventMessage
- thycotic-ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.EventPipelines.PipelinePolicyProcessConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelinePoliciesProcessBlockingMessage
- thycotic-ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.EventPipelines.PipelinePolicyProcessConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelinePoliciesProcessMessage
- thycotic-ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.EventPipelines.PipelineProcessConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelineProcessBlockingMessageWithPolicies
- thycotic-ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.EventPipelines.PipelineProcessConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelineProcessMessageWithPolicies
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.EventPipelines.PipelinePolicyProcessEventConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelinePolicyProcessEventBlockingMessage

- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.EventPipelines.PipelinePolicyProcessEventConsumer:Thycotic.Messages.ihawu.Areas.EventPipelines.Request.PipelinePolicyProcessEventMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.EventPipelines.PipelineProcessScheduledEventConsumer:Thycotic.ihawu.Business.Messages.Areas.EventPipelines.ProcessPipelineScheduledEventMessage

Heartbeat and Remote Password Change

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.CheckinExpiredCheckedoutSecretConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.CheckinExpiredCheckedoutSecretMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ExpiredSecretLocalPasswordChangeConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.ExpiredSecretLocalPasswordChangeMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ExpiredSecretLocalPasswordChangeConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.RunNowExpiredSecretLocalPasswordChangeMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ExpiredSecretPasswordChangeConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.ExpiredSecretPasswordChangeMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ExpiredSecretPasswordChangeConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.RunNowExpiredSecretPasswordChangeMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ProcessHeartbeatConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.ProcessHeartbeatMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ProcessHeartbeatConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.RunNowProcessHeartbeatMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ProcessLocalHeartbeatConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.ProcessLocalHeartbeatMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RemotePasswordChanging.ProcessLocalHeartbeatConsumer:Thycotic.ihawu.Business.Messages.Areas.RemotePasswordChanging.Request.RunNowProcessLocalHeartbeatMessage

Import

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.Import.SecretImportConsumer:Thycotic.ihawu.Business.Messages.Import.SecretImportBulkMessage
- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.Import.SecretImportFileConsumer:Thycotic.ihawu.Business.Messages.Import.SecretImportFileMessage

Management: Backup, and Cleanup

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.OnPremisesOnly.BackgroundWorkerTaskConsumer:Thycotic.ihawu.Business.Messages.Areas.OnPremisesOnly.BackgroundWorkerTaskMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.OnPremisesOnly.BackupConsumer:Thycotic.ihawu.Business.Messages.Areas.OnPremisesOnly.BackupMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.OnPremisesOnly.BackupConsumer:Thycotic.ihawu.Business.Messages.Areas.OnPremisesOnly.RunNowBackupMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.OnPremisesOnly.GenerateSLMConsumer:Thycotic.ihawu.Business.Logic.Areas.OnPremisesOnly.GenerateSLMMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.SessionArchiving.RecordedSessionsArchiveConsumer:Thycotic.ihawu.Business.Messages.Areas.SessionArchiving.Request.ArchiveRecordedSessionsMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.SessionArchiving.RecordedSessionsArchiveConsumer:Thycotic.ihawu.Business.Messages.Areas.SessionArchiving.Request.DeleteRecordedSessionsMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.SessionArchiving.RecordedSessionsArchiveConsumer:Thycotic.ihawu.Business.Messages.Areas.SessionArchiving.Request.RunNowDeleteRecordedSessionsMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.TruncateRecords.TruncateRecordsConsumer:Thycotic.ihawu.Business.Messages.Areas.TruncateRecords.TruncateRecordsForAllConfigurationsMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.TruncateRecords.TruncateRecordsConsumer:Thycotic.ihawu.Business.Messages.Areas.TruncateRecords.TruncateRecordsForConfigurationMessage
- thycotic-ss:Thycotic.MessageQueue.Common.Consumers.AutomaticSink.CreateAutomaticSinkConsumer:Thycotic.MessageQueue.Common.Messages.AutomaticSink.Request.CreateAutomaticSinkMessage

Distributed Engine Management

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.DistributedEngine.EngineStatusUpdateConsumer:Thycotic.ihawu.Business.Messages.Areas.DistributedEngine.Request.EngineStatusUpdateMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.DistributedEngine.TruncateEngineLogConsumer:Thycotic.ihawu.Business.Messages.Areas.DistributedEngine.Request.TruncateEngineLogMessage

Password Generation

- thycotic-

SecretServer:Thycotic.ihawu.Business.Logic.Areas.PasswordGeneration.GeneratePasswordConsumer:Thycotic.ihawu.Business.Messages.Areas.PasswordGeneration.Request.GeneratePasswordMessage

Reports

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.Report.EmailReportConsumer:Thycotic.Messages.ihawu.Areas.Email.Request.EmailReportMessage-SecretServer:
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Report.ScheduledReportConsumer:Thycotic.ihawu.Business.Messages.Areas.Reports.Request.ProcessReportsMessage

Run Now

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessDashboardJsonValidationConsumer:Thycotic.ihawu.Business.Messages.Areas.RunOnceTasks.RunNowProcessDashboardJsonValidationMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessFieldEncryptionChangesConsumer:Thycotic.ihawu.Business.Messages.Areas.RunOnceTasks.RunNowProcessFieldEncryptionChangesMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessFieldEncryptionChangesConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.ProcessFieldEncryptionChangesMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessSecretKeyRotationConsumer:Thycotic.ihawu.Business.Logic.Areas.SecretKeyRotation.Messages.RunNowProcessSecretKeyRotationMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessSecretPolicyChangesConsumer:Thycotic.ihawu.Business.Messages.Areas.RunOnceTasks.RunNowProcessSecretPolicyChangesMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowProcessSecretPolicyChangesConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.ProcessSecretPolicyChangesMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.RunOnceTasks.RunNowToggleHsmConsumer:Thycotic.ihawu.Business.Logic.Areas.SecretKeyRotation.Messages.RunNowToggleHsmMessage

Scheduled Tasks

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.DatabaseCleanupConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.DatabaseCleanupMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.EventQueueMonitorConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.EventQueueMessage
- thycotic-

ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.ExpiringLicenseTaskConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.ExpiringLicenseTaskMessage

- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.ExpiringSecretTaskConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.ExpiringSecretTaskMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.PasswordRequirementConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.PasswordRequirementMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.SqlReplicationConflictConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.SqlReplicationConflictMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ScheduledTask.TruncateDatabaseCacheConsumer:Thycotic.ihawu.Business.Messages.Areas.ScheduledTask.Request.TruncateDatabaseCacheMessage

Search

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.Search.ProxySessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.ProxySessionDataHashIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.ProxySessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.ProxySessionDataHashReIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.ProxySessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RunNowProxySessionDataHashReIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.RdpSessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RdpSessionDataHashIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.RdpSessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RdpSessionDataHashReIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.RdpSessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RunNowRdpSessionDataHashReIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.SecretItemHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RunNowSecretItemHashIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.SecretItemHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.SecretItemHashIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.SessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.RunNowSessionDataHashReIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.SessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.SessionDataHashIndexRequest
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Search.SessionDataHashConsumer:Thycotic.ihawu.Business.Messages.Areas.Search.Request.SessionDataHashReIndexRequest

SSH Terminal

- thycotic-SecretServer:Thycotic.ihawu.BackgroundWorker.Logic.Areas.SSHTerminal.TerminalCommandBackgroundConsumer:Thycotic.Messages.DE.Server.Areas.SSHTerminal.Request.TerminalCommandMessage

Delinea Privilege Behavior Analytics Integration

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaAppendMetadataSinkConsumer:Thycotic.Messages.SA.Areas.Metadata.Request.SAAppendMetadataSinkMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaCreateMetadataSinkConsumer:Thycotic.Messages.SA.Areas.Metadata.Request.SACreateMetadataSinkMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaDirectiveConsumer:Thycotic.ihawu.Business.Messages.Areas.PBA.Request.DirectiveProcessMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaEventConsumer:Thycotic.Messages.SA.Areas.EventData.Request.SAEventMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaEventUploadConsumer:Thycotic.Messages.SA.Areas.EventData.Request.SAEventUploadMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.Pba.PbaMetadataUploadConsumer:Thycotic.Messages.SA.Areas.Metadata.Request.SAMetadataUploadMessage
- thycotic-ss:Thycotic.SecurityAnalytics.DataUploader.Consumers.DirectiveAddConsumer:Thycotic.Messages.SA.Areas.Directive.Request.SADirectiveSendMessage
- thycotic-ss:Thycotic.SecurityAnalytics.DataUploader.Consumers.DirectiveCheckConsumer:Thycotic.Messages.SA.Areas.Directive.Request.SADirectiveCheckMessage
- thycotic-ss:Thycotic.SecurityAnalytics.DataUploader.Consumers.HealthCheckConsumer:Thycotic.Messages.SA.Areas.Status.Request.SAHealthCheckMessage
- thycotic-ss:Thycotic.SecurityAnalytics.DataUploader.Consumers.HeartbeatConsumer:Thycotic.Messages.SA.Areas.Status.Request.SAHeartbeatMessage

Delinea Privilege Manager Integration

- thycotic-SecretServer:Thycotic.ihawu.BackgroundWorker.Logic.Areas.TmsNotifications.NotifyTmsDatabaseUpdatedConsumer:Thycotic.Messages.ihawu.Areas.TmsNotifications.Request.NotifyTmsDatabaseUpdatedMessage
- thycotic-ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.TmsNotifications.NotifyTmsEmailSettingsUpdatedConsumer:Thycotic.Messages.ihawu.Areas.TmsNotifications.Request.NotifyTmsEmailSettingsUpdatedMessage
- thycotic-ss:Thycotic.ihawu.BackgroundWorker.Logic.Areas.TmsNotifications.NotifyTmsLicenseUpdatedConsumer:Thycotic.Messages.ihawu.Areas.TmsNotifications.Request.NotifyTmsLicenseUpdatedMessage

Delinea Telemetry

- thycotic-SecretServer:Thycotic.ihawu.BackgroundWorker.Logic.Areas.Telemetry.TelemetryConsumer:Thycotic.Messages.ihawu.Areas.Telemetry.Request.SendAnonymousTelemetryMessage

Delinea One Identify Provider Integration

- thycotic-SecretServer:Thycotic.ihawu.Business.Logic.Areas.ThycoticOne.ThycoticOneSyncUserConsumer:Thycotic.ihawu.Business.Messages.Areas.ThycoticOne.Request.ThycoticOneScheduledSyncMessage
- thycotic-ss:Thycotic.ihawu.Business.Logic.Areas.ThycoticOne.ThycoticOneSyncUserConsumer:Thycotic.ihawu.Business.Messages.Areas.ThycoticOne.Request.ThycoticOneSyncUserMessage

Engine Role Queues

List of queues for engines' functional areas.

Note: In the example listed below, the Secret Server site name is called "Gamma-Engines".

Active Directory Synchronization

- Gamma-Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.ADSyncRequestConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.ADSyncMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.AllUsersByDomainQueryConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.AllUsersByDomainQueryMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.GenericQueryConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.GroupsAndMembersQueryMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.GenericQueryConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.GroupsByDomainQueryMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.GenericQueryConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.UsersByGroupsQueryMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.ResolveDomainNameConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.ResolveDomainDistinguishedNameMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.AdSync.Areas.ActiveDirectory.ResolveDomainNameConsumer:Thycotic.Messages.DE.Engine.Areas.ActiveDirectory.Request.ResolveFullyQualifiedDomainNameMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.AdSync.Areas.Authentication.AuthenticateByAdConsumer:Thycotic.Messages.DE.Engine.Areas.Authenticate.Request.AuthenticateByAdMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.AdSync.Areas.General.DomainCredentialTestConsumer:Thycotic.Messages.DE.Engine.Areas.General.Request.DomainCredentialTestMessage

Discovery

- Gamma-Engines:Thycotic.DE.Feature.SS.LocalAccountDiscovery.Areas.Discovery.HostRangeConsumer:Thycotic.Messages.DE.Engine.Areas.Discovery.Request.ScanHostRangeMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.LocalAccountDiscovery.Areas.Discovery.HostRangeConsumer:Thycotic.Messages.DE.Engine.Areas.Discovery.Request.SpecificOuScanHostRangeMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.LocalAccountDiscovery.Areas.Discovery.LocalAccountConsumer:Thycotic.Messages.DE.Engine.Areas.Discovery.Request.ScanLocalAccountMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.LocalAccountDiscovery.Areas.Discovery.MachineConsumer:Thycotic.Messages.DE.Engine.Areas.Discovery.Request.ScanMachineMessage

Heartbeat, Remote Password Change, and Dependency

- Gamma-Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.BlockingChangePasswordConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request.BlockingPasswordChangeMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.BlockingPrivilegeChangePasswordConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request.BlockingPrivilegedPasswordChangeMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.Heartbeat.SecretHeartbeatConsumer:Thycotic.Messages.DE.Engine.Areas.Heartbeat.Request.SecretHeartbeatMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.SecretBasicChangePasswordConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request.SecretBasicPasswordChangeMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.SecretPrivilegeChangePasswordConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request.SecretPrivilegedPasswordChangeMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.SecretRunDependenciesConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request.SecretChangeDependencyMessage Gamma-Engines:Thycotic.DE.Feature.SS.PasswordChanging.Areas.Verification.VerifyPasswordConsumer:Thycotic.Messages.DE.Engine.Areas.Verify.Request.VerifyPasswordMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.ServiceAccountManagement.Areas.Dependency.DependencyConsumer:Thycotic.Messages.DE.Engine.Areas.Discovery.Request.ScanDependencyMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.ServiceAccountManagement.Areas.Dependency.SecretTestDependencyConsumer:Thycotic.Messages.DE.Engine.Areas.PasswordChanging.Request.SecretTestDependencyMessage

Management

- Gamma-Engines:Thycotic.DistributedEngine.Logic.Areas.Connectivity.PingConsumer:Thycotic.Messages.DE.Engine.Areas.Connectivity.Request.PingMessage

- Gamma-Engines:Thycotic.MessageQueue.Common.Consumers.AutomaticSink.CreateAutomaticSinkConsumer:Thycotic.MessageQueue.Common.Messages.AutomaticSink.Request.CreateAutomaticSinkMessage

Proxy

- Gamma-Engines:Thycotic.DE.Feature.SS.RdpProxy.AssignProxiedRdpSessionConsumer:Thycotic.Messages.DE.Engine.Areas.RDPProxy.Request.AssignProxiedRdpSessionMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.SshProxy.Areas.Proxy.AssignProxiedSessionConsumer:Thycotic.Messages.DE.Engine.Areas.SSHProxy.Request.AssignProxiedSessionMessage

Scripting

- Gamma-Engines:Thycotic.DistributedEngine.Logic.Areas.Script.ScriptConsumer:Thycotic.Messages.DE.Engine.Areas.Script.Request.PowerShellScriptMessage
- Gamma-Engines:Thycotic.DistributedEngine.Logic.Areas.Script.ScriptConsumer:Thycotic.Messages.DE.Engine.Areas.Script.Request.SqlScriptMessage
- Gamma-Engines:Thycotic.DistributedEngine.Logic.Areas.Script.ScriptConsumer:Thycotic.Messages.DE.Engine.Areas.Script.Request.SshScriptMessage

Syslog Integration

- Gamma-Engines:Thycotic.DE.Feature.SS.AdvancedAuditing.Areas.SIEM.SysLogConsumer:Thycotic.Messages.DE.Engine.Areas.SIEM.Request.SysLogMessage

Delinea Privilege Behavior Analytics Integration

- Gamma-Engines:Thycotic.DE.Feature.SS.Pba.Areas.Event.PbaEventConsumer:Thycotic.Messages.SA.Areas.EventData.Request.SAEventMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.Pba.Areas.MetaData.PbaAppendMetadataSinkConsumer:Thycotic.Messages.SA.Areas.Metadata.Request.SAAppendMetadataSinkMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.Pba.Areas.MetaData.PbaCreateMetadataSinkConsumer:Thycotic.Messages.SA.Areas.Metadata.Request.SACreateMetadataSinkMessage
- Gamma-Engines:Thycotic.SecurityAnalytics.DataUploader.Consumers.DirectiveAddConsumer:Thycotic.Messages.SA.Areas.Directive.Request.SADirectiveSendMessage
- Gamma-Engines:Thycotic.SecurityAnalytics.DataUploader.Consumers.DirectiveCheckConsumer:Thycotic.Messages.SA.Areas.Directive.Request.SADirectiveCheckMessage
- Gamma-Engines:Thycotic.SecurityAnalytics.DataUploader.Consumers.HealthCheckConsumer:Thycotic.Messages.SA.Areas.Status.Request.SAHealthCheckMessage
- Gamma-Engines:Thycotic.SecurityAnalytics.DataUploader.Consumers.HeartbeatConsumer:Thycotic.Messages.SA.Areas.Status.Request.SAHeartbeatMessage

Ticketing System Integration

- Gamma-Engines:Thycotic.DE.Feature.SS.SecretWorkflow.Areas.TicketingSystem.TicketingAddCommentConsumer:Thycotic.Messages.DE.Engine.Areas.TicketingSystem.Request.TicketingAddCommentBasicRequest
- Gamma-Engines:Thycotic.DE.Feature.SS.SecretWorkflow.Areas.TicketingSystem.TicketingAddCommentConsumer:Thycotic.Messages.DE.Engine.Areas.TicketingSystem.Request.TicketingAddCommentMessage
- Gamma-Engines:Thycotic.DE.Feature.SS.SecretWorkflow.Areas.TicketingSystem.TicketingGetStatusConsumer:Thycotic.Messages.DE.Engine.Areas.TicketingSystem.Request.TicketingGetStatusMessage

Engine Worker Role Queues

List of queues for engine worker's functional areas:

Active Directory Synchronization

- thycotic-SecretServer-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.ActiveDirectory.ActiveDirectorySynchronizationConsumer:Thycotic.Messages.DE.Server.Areas.ActiveDirectory.Request.ADSyncMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.ActiveDirectory.AllUsersByDomainQueryConsumer:Thycotic.Messages.DE.Server.Areas.ActiveDirectory.Request.AllUsersByDomainQueryMessage

Discovery

- thycotic-SecretServer-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.Discovery.ScanDepedencyConsumer:Thycotic.Messages.DE.Server.Areas.Discovery.Request.ScanDependencyMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.Discovery.ScanHostRangeResponseConsumer:Thycotic.Messages.DE.Server.Areas.Discovery.Request.ScanHostRangeMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.Discovery.ScanLocalAccountConsumer:Thycotic.Messages.DE.Server.Areas.Discovery.Request.ScanLocalAccountMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.Discovery.ScanMachineResponseConsumer:Thycotic.Messages.DE.Server.Areas.Discovery.Request.ScanMachineMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.Discovery.SpecificOuScanHostRangeResponseConsumer:Thycotic.Messages.DE.Server.Areas.Discovery.Request.SpecificOuScanHostRangeMessage

RDP Proxy, SSH Proxy, and SSH Terminal

- thycotic-SecretServer-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.RDPProxy.AppendKeystrokeDataConsumer:Thycotic.Messages.DE.Engine.Areas.RDPProxy.Request.AppendKeystrokeDataMessage

- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.AppendSessionDataConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.AppendSessionDataMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.CloseSecretSessionConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.EndSessionDataMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.EndRdpProxySessionConsumer:Thycotic.Messages.DE.Engine.Areas.RDPProxy.Request.EndRdpProxySessionMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.GetStatusUpdatesRequestConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.GetStatusUpdatesMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.InitiateRDPProxiedSessionConsumer:Thycotic.Messages.DE.Engine.Areas.RDPProxy.Request.InitiateProxiedRdpSessionMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.InitiateSSHProxiedSessionConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.InitiateProxiedSessionMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.InitiateSshSessionDataCaptureSinkConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.InitiateProxiedSessionDataCaptureSinkMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.RdpProxySessionStatusesConsumer:Thycotic.Messages.DE.Engine.Areas.RDPProxy.Request.GetRdpProxySessionStatusesMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.UpdateSessionsRequestConsumer:Thycotic.Messages.DE.Server.Areas.SSHProxy.Request.UpdateSessionsMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHProxy.UpdateUserPasswordRequestConsumer:Thycotic.Messages.DE.Server.Areas.SSHTerminal.Request.UpdateUserPasswordMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SSHTerminal.TerminalCommandEngineConsumer:Thycotic.Messages.DE.Server.Areas.SSHTerminal.Request.TerminalCommandMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.UserSession.CloseUserSessionConsumer:Thycotic.Messages.DE.Server.Areas.UserSession.CloseUserSessionMessage

Syslog Integration

- thycotic-SecretServer-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.SEIM.SysLogResultResponseConsumer:Thycotic.Messages.DE.Server.Areas.SEIM.Request.SysLogResultMessage

Heartbeat, Remote Password Change, and Dependency

- thycotic-SecretServer-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.Dependency.DependencyChangeConsumer:Thycotic.Messages.DE.Server.Areas.Dependency.Request.DependencyChangeMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.Heartbeat.SecretHeartbeatConsumer:Thycotic.Messages.DE.Server.Areas.Heartbeat.Request.SecretHeartbeatMessage

- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.PasswordChanging.RemotePasswordChangeResponseStoreConsumer:Thycotic.Messages.DE.Server.Areas.PasswordChanging.Request.RemotePasswordChangeMessage

Delinea Privilege Behavior Analytics Integration

- thycotic-SecretServer-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.PasswordChanging.PbaDisableConsumer:Thycotic.Messages.DE.Server.Areas.PBA.PbaDisableMessage

Distributed Engine Management

- thycotic-SecretServer-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.Connectivity.PingConsumer:Thycotic.Messages.DE.Server.Areas.Connectivity.Request.PingMessage
- thycotic-ss-engine-response:Thycotic.ihawu.EngineWorker.Logic.Areas.Maintenance.LogConsumer:Thycotic.Messages.DE.Server.Areas.Maintenance.Request.EngineLogMessage
- thycotic-ss-engine-response:Thycotic.MessageQueue.Common.Consumers.AutomaticSink.CreateAutomaticSinkConsumer:Thycotic.MessageQueue.Common.Messages.AutomaticSink.Request.CreateAutomaticSinkMessage

Session Recording Worker

List of queues for session recording worker's functional areas:

Post Recording

- thycotic-sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostMetadataConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.ProcessUploadedMetadataMessage
- thycotic-sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostRecordedSessionConsumer:Thycotic.Messages.DE.Server.Areas.AdvancedSessionRecording.Request.RecordedSessionChunkMessage
- thycotic-sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostRecordedSessionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.ProcessBusStreamedSessionMessage
- thycotic-sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostRecordedSessionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.ProcessUploadedSessionMessage

Video Conversion

- thycotic-sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.ConvertAllVideosMessage
- thycotic-

sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.ConvertVideoMessage

- thycotic-sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.DeleteOldCompletedImagesMessage
- thycotic-sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.RunNowConvertVideoMessage
- thycotic-sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.RunNowSetStatusForTimedOutSessionsMessage
- thycotic-sessionrec:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.VideoConversionConsumer:Thycotic.Messages.ihawu.Areas.SessionRecording.Request.SetStatusForTimedOutSessionsMessage

Post Recording (Legacy)

- thycotic-sr-agent-response:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostMetadataConsumer:Thycotic.Messages.DE.Server.Areas.AdvancedSessionRecording.Request.PostMetadataMessage
- thycotic-sr-agent-response:Thycotic.ihawu.SessionRecordingWorker.Logic.Areas.SessionRecording.PostRecordedSessionConsumer:Thycotic.Messages.DE.Server.Areas.AdvancedSessionRecording.Request.PostRecordedSessionMessage

Management

- thycotic-sessionrec:Thycotic.MessageQueue.Common.Consumers.AutomaticSink.CreateAutomaticSinkConsumer:Thycotic.MessageQueue.Common.Messages.AutomaticSink.Request.CreateAutomaticSinkMessage

RDP Proxy

The RDP Proxy feature in Secret Server enhances security by routing Remote Desktop Protocol (RDP) connections through Secret Server, ensuring that secret credentials are protected during remote access sessions. This proxying mechanism can be configured in two ways: the recommended method, which uses temporary credentials to connect to the RDP proxy and then to the remote server, and an alternative method that tunnels the RDP connection through an SSH proxy. The recommended method is preferred as it prevents secret credentials from reaching the client machine, thereby enhancing security. The RDP Proxy supports up to 100 concurrent connections and ensures that all RDP traffic is securely monitored and recorded if session recording is enabled. This setup mitigates risks associated with direct RDP access and helps prevent lateral movement of malware between machines.

RDP Proxy Configuration

RDP Proxy requires .NET 4.7.2 or later. Also, CredSSP, NTLMv2, and NLA must be enabled. LAN Manager authentication must be set to NTLMv2 (Lmcompatibilitylevel configured to 3 or above). See the RDP Proxy Configuration for details.

Overview

The RDP Proxying feature allows RDP connections, established using a launcher, to be routed through Secret Server. You can set it up one of two ways:

- Recommended method: The launcher connects to the newer RDP proxy with temporary credentials, and the RDP proxy connects to the remote server using the protected credentials from the secret. This method is preferred because it prevents the secret credentials from reaching the client machine. For this method, you simply configure the RDP proxy.
- Alternative method: The launcher uses an SSH proxy to tunnel a local RDP connection to a remote server. This method does not protect the credential from reaching the client machine. For this method you configure the SSH proxy and enable SSH tunneling.

We provide the alternate method to support legacy installations and troubleshooting (it can potentially be more stable when the RDP proxy does not work).

These two approaches to RDP proxying are not compatible—you may use one or the other but not both. We performance tested both methods. Either can support 100 concurrent connections.

Recommended Method

How It Works

1. The user clicks the RDP launcher in Secret Server.
2. The launcher executes on the client's machine.
3. The launcher establishes a connection to the RDP Proxy using credentials generated for the session.

   These credentials are short-lived and can only be used within a 15-minute window. To support reconnects in keeping with the RDP protocol, the window resets upon reconnect.

4. Once the launcher has successfully authenticated with the RDP proxy, the RDP proxy looks up the credentials and target hostname to connect to.

   The secret credentials *do not* get served to the client machine in this flow, which improves credential security.

5. The RDP proxy connects to the desired remote host with the secret credentials.
6. The RDP session is established.

7. RDP traffic is sent back and forth over the RDP proxy, session keystrokes are monitored if session recording is enabled.

Configuration

1. Navigate to the **Admin > Proxying** page.
2. Click the **RDP Proxy** tab.
2. If necessary, enable the RDP proxy.
3. Click the **Endpoints** tab to ensure that your server nodes, sites, and engines have RDP Proxy enabled.
4. Proxied RDP secrets now launch into the RDP proxy using short-lived credentials, protecting the secret credentials from the client machine.

Configuration Settings

The RDP proxy configuration settings for the recommended method:

- **Enable RDP Proxy:** This setting determines whether or not the RDP proxy is enabled
- **RDP Proxy Port:** This setting is the port that the RDP proxy runs on (defaulting to 3390). You usually cannot set this to 3389 as that port is already occupied by default by the Windows operating system.
- **Validate Remote Certificates:**Delinea recommends that you operate in an environment where RDP server certificates are created by a controlled CA and are trusted by machines in the domain. If that is not possible, you can disable remote certificate validation to allow connection to machines that do not serve trusted certificates. See Remote Certificate Validation for more details.
- **Allow AD site selection:** This setting allows you to select any configured sites when using the RDP launcher on an Active Directory secret. This allows a secret credential to access machines that may exist in different network boundaries.
- **Proxy New Secrets By Default:** This setting determines if SSH and RDP secrets are created with "Proxy Enabled" set by default. This setting is shared with the SSH proxy configuration.
- **Days To Keep Operational Logs:** This setting determines how long, in days, the operational logs for the RDP proxy are kept.
- **RDP Server Certificate:** This setting is the certificate that is served to the clients who connect to the RDP proxy. You can generate a certificate for a given DNS name, or you can upload your own. Please see Microsoft's documentation for instructions on creating Certificates for more details on how to create a certificate via a Windows Trusted Domain.
- **Use Secret Server RDP Client:** When using protocol handler and this is disabled, the RDP window label only displays the proxy server name. When this setting is enabled, the RDP window label also displays the target server and that the connection is proxied. This setting is found under the launcher itself in **Admin > Secret Templates > Launchers**.

Alternative Method

This approach is not recommended as it exposes the secret credentials to the client machine.

How It Works

1. The user clicks the RDP launcher in Secret Server.
2. The launcher executes on the client's machine.
3. The launcher establishes a connection to the SSH proxy to begin port forwarding.
4. The launcher authenticates with the SSH Proxy.
5. The launcher opens a socket.
6. The launcher listens for a connection on an available ephemeral port (the forwarding port) on the client's machine.
7. RDP launches on the client machine using the secret credentials and connects locally to the forwarding port.
8. All RDP traffic for this session is routed through the SSH tunnel to Secret Server, then forwarded to the target machine.
9. The RDP session is established.

Configuration

1. Navigate to the **Admin > Proxying** page.
2. Enable the **Enable SSH Tunneling** option.
3. Click the **Endpoints** tab to ensure that your server nodes, sites, and engines are properly configured.
4. Proxied RDP secrets now launch into the SSH proxy using local port forwarding.

Known Issues

"Could not load file or assembly..." Error

Error appears in Secret Server.log or DE.log. Install the most recent version of the .NET Framework to correct it.

RDP Proxy Does Not Work with FIPS Validation

RDP proxy does not work on machines the have the FIPS validation security policy active. No fix is currently available.

Using Remote Certificate Validation with Secret Server RDP Proxy

Configure Certificate from Trusted Source

The most correct and complete way of configuring the certificate is replacing the RDP certificate with a certificate signed by a trusted certificate authority. There are various ways to achieve this by for example manually generating certificates from a trusted source and configuring these certificates on the target machines on the RDP listener by utilizing the following command:

```
wmic /namespace:\\root\cimv2\TerminalServices PATH Win32_TSGeneralSetting Set
SSLCertificateSHA1Hash="<thumbprint of the certificate>"
```

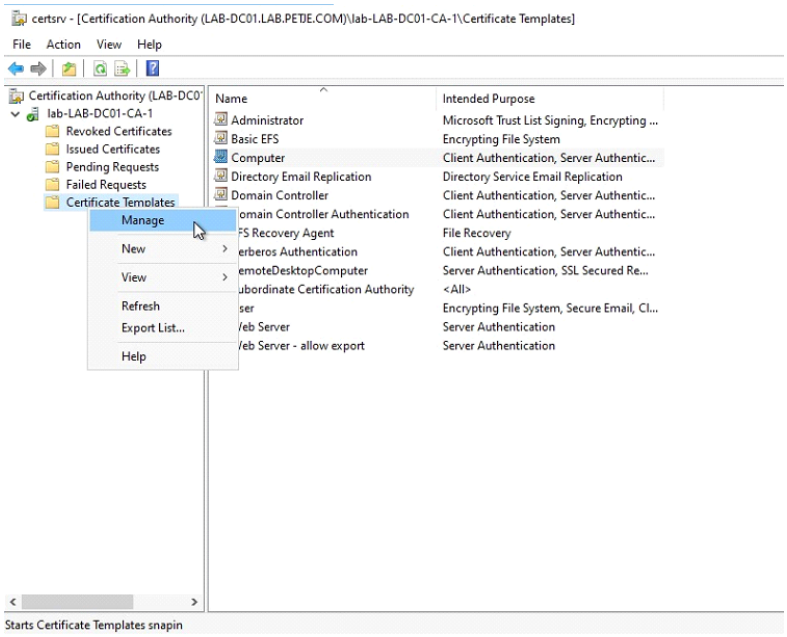
> Note that the certificates need to have at least the following key usage policy configured: Server Authentication (1.3.6.1.5.5.7.3.1)

A more efficient way is to configure an Active Directory Certificate Authority, which will issue the respective certificates combined with GPO settings to reconfigure the RDS listener to utilize that was issued by the Certificate Authority. The steps to configure a Windows CA and GPO are outlined below.

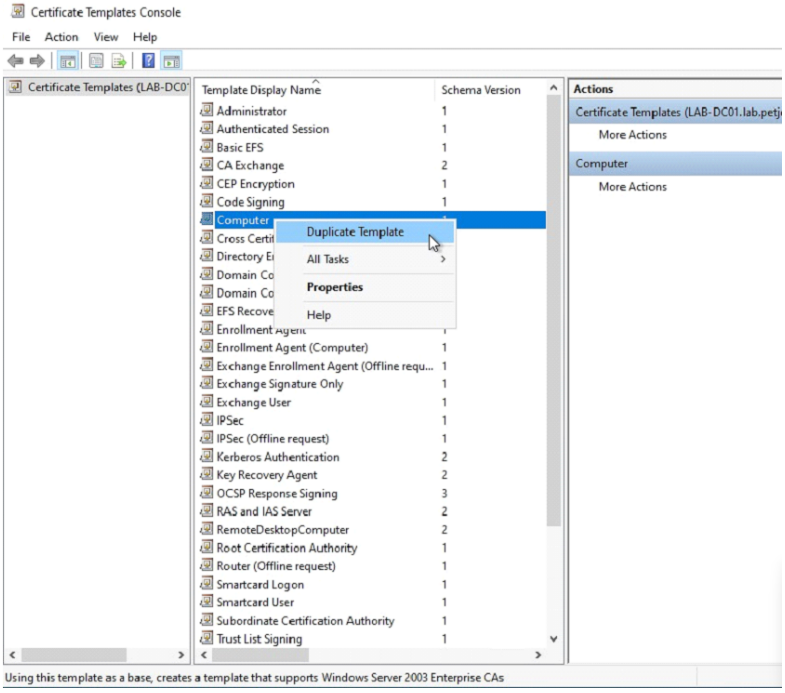
Configure Windows Certificate Authority Configuration

Base requirement for deploying RDP certificates is to have a configured and functioning Microsoft Certificate Authority. Configuration of this is outside of the scope of this document.

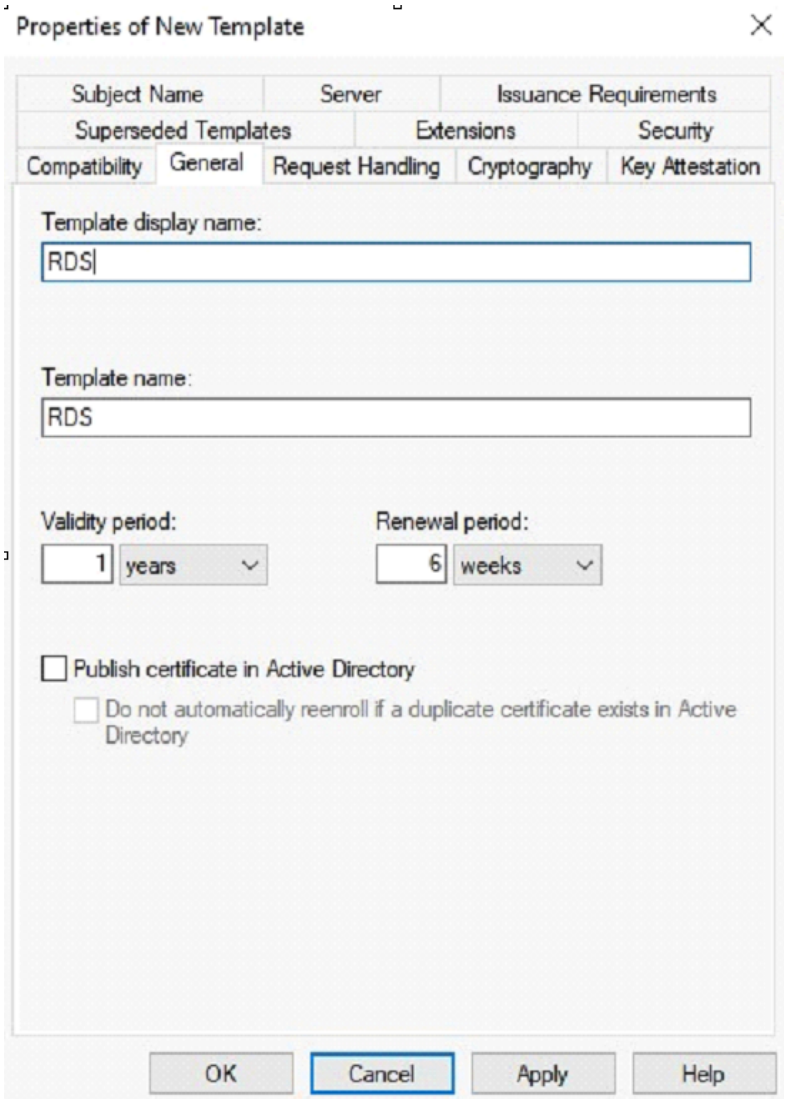
1. First step is to prepare a certificate template which will be used to issue RDP Certificates. To do so, open the Certificate Manager and navigate to the Management of Certificate Templates.



2. In the Certificate Templates Console, select the Computer template and duplicate this template.

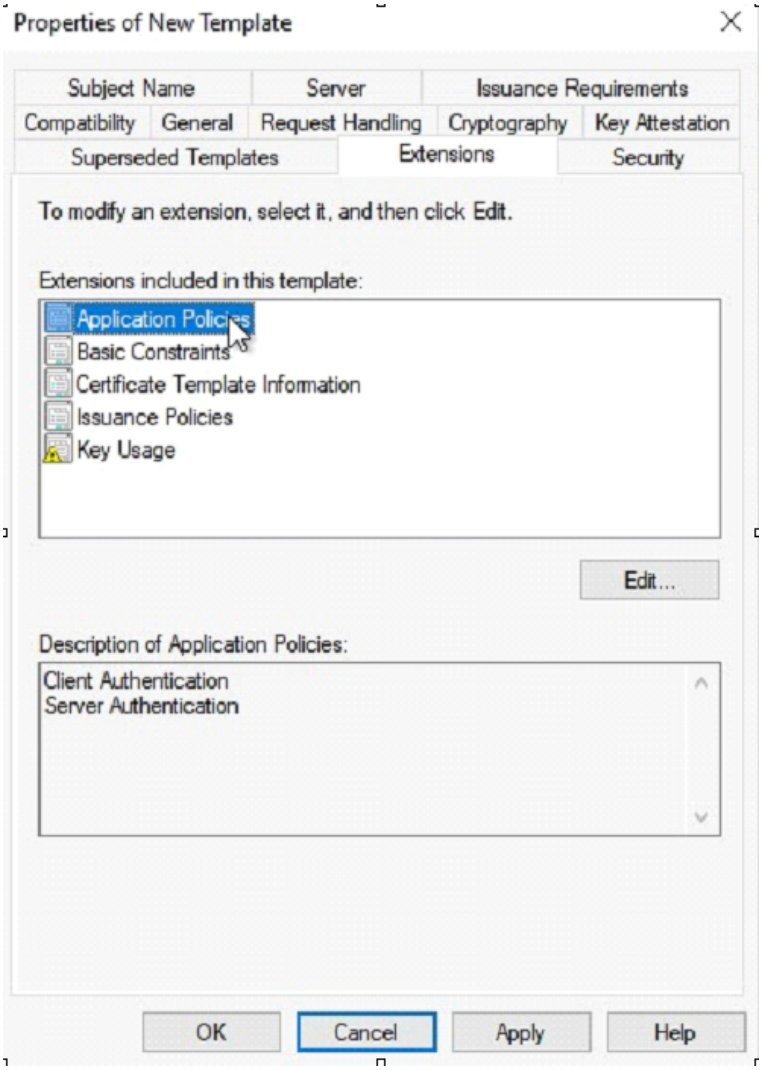


3. Provide a new name for the template. In the example below the name is RDS (any name can be provided). Depending on the company requirements, the validity period and renewal period can be adjusted. This however is not mandatory.

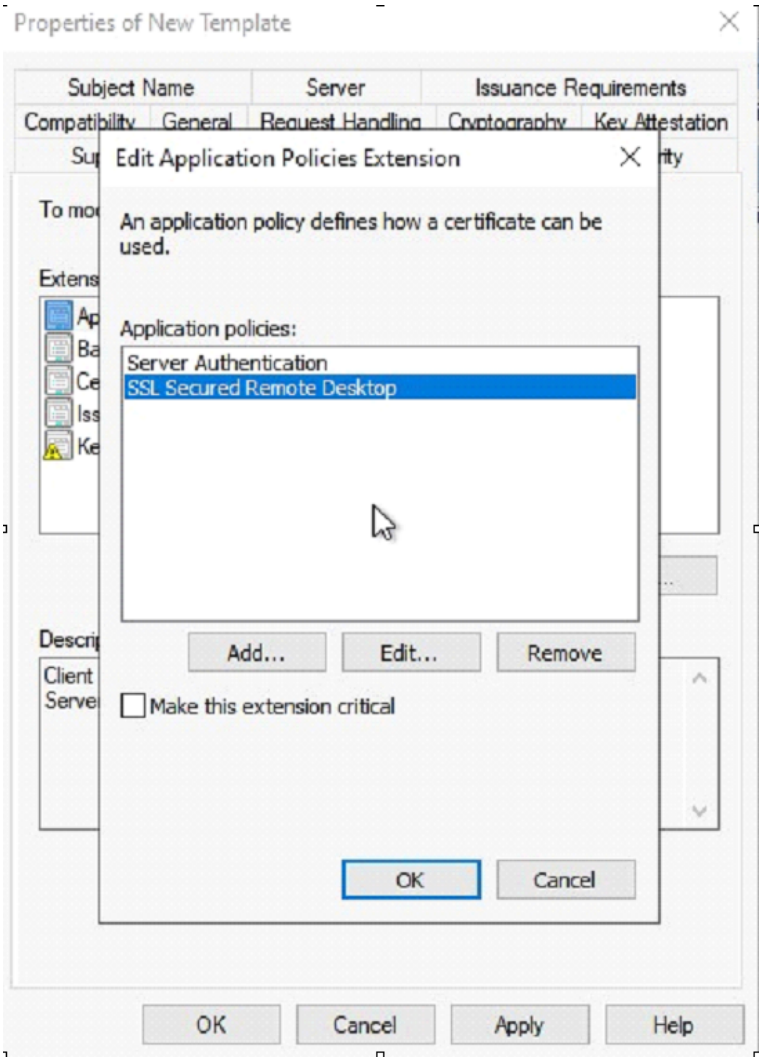


4. Next, configuration of the usage of the certificate needs to be adjusted.

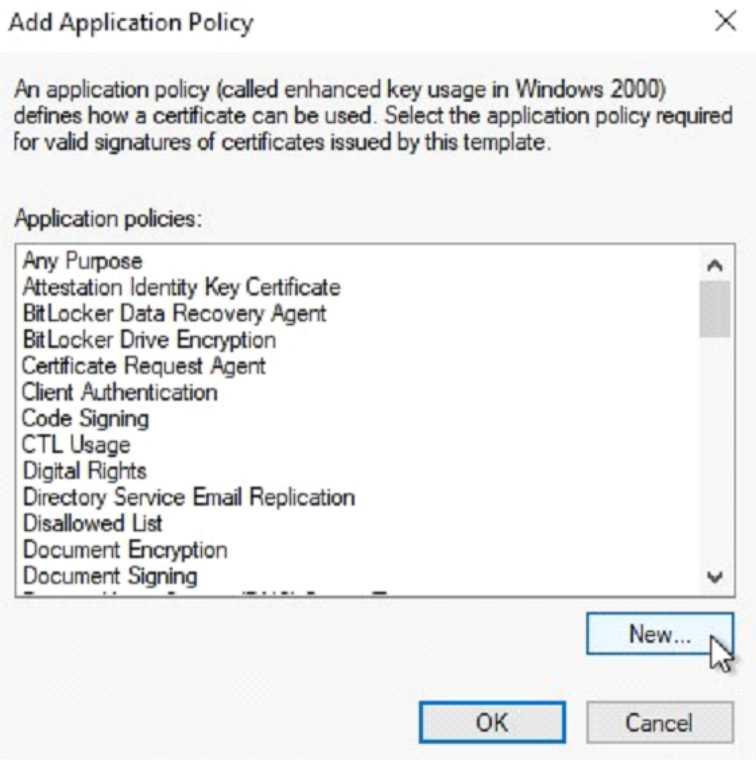
- Navigate to the Extension Tab and edit the application policies.



- Make sure to configure the application policies to contain `Server Authentication` and `SSL Secured Remote Desktop`. Depending of the version of Windows and configuration it is possible the SSL Secured Remote Desktop is not available. In that case, create a new application policy.



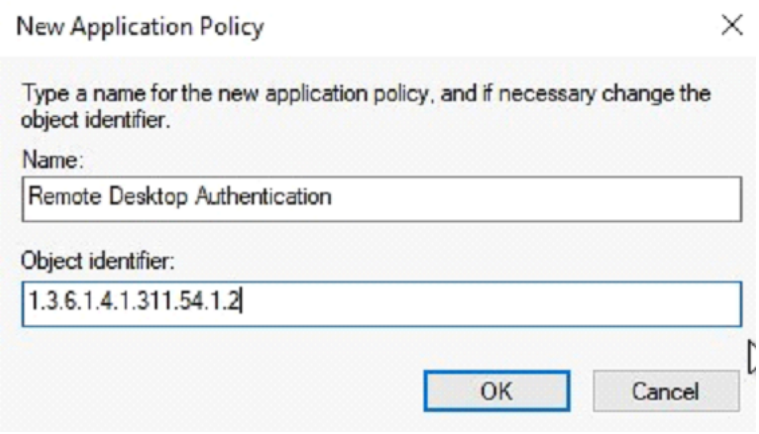
- When adding the application policy, a list with configured policies is being showed, and the correct policies can be added as per previous step.



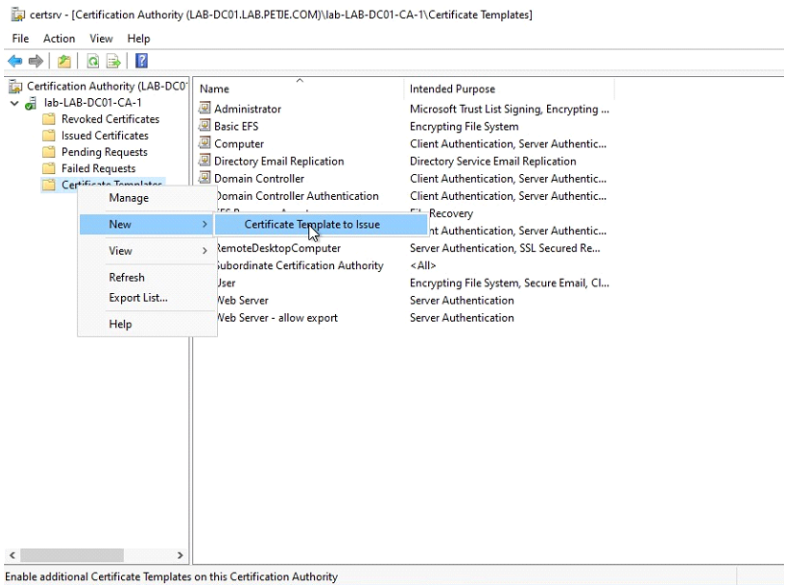
- When indeed the SSL Secured Remote Desktop is not available, it can be created by pressing the New button, with the following details. The actual name is not really relevant, just make sure to name it logically as it will show on the certificate.

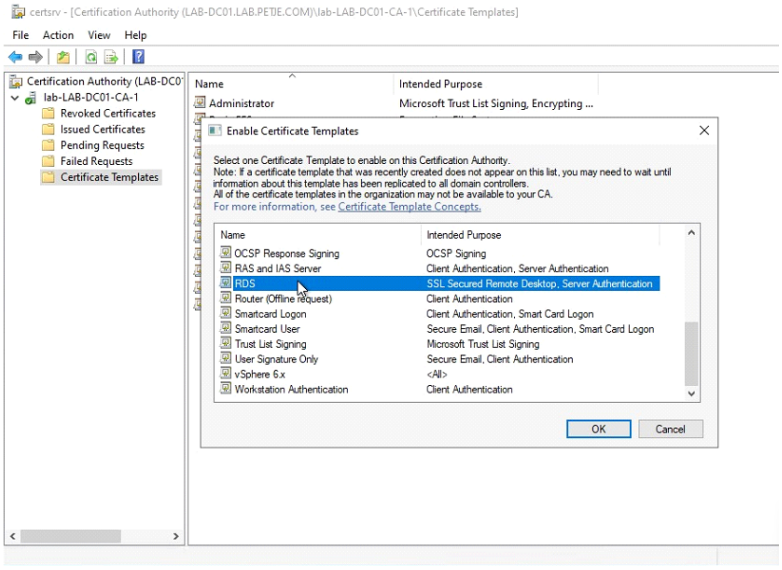
  Name = `Remote Desktop Authentication`

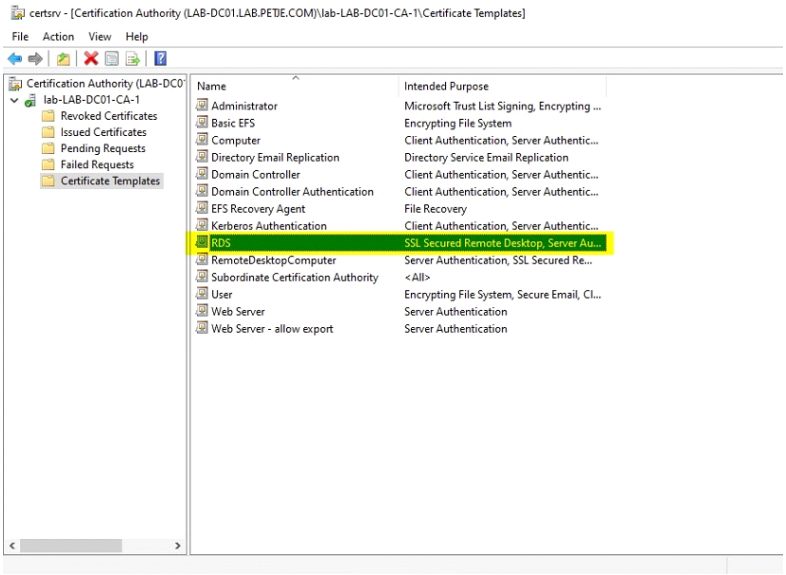
  Object Identifier = `1.3.6.1.4.1.311.54.1.2`



5. When the certificate template has been created it needs to be published on the Certificate Server.


certsrv - [Certification Authority (LAB-DC01.LAB.PETJE.COM)\lab-LAB-DC01-CA-1\Certificate Templates]
File Action View Help
Certification Authority (LAB-DC0
lab-LAB-DC01-CA-1
Revoked Certificates
Issued Certificates
Pending Requests
Failed Requests
Manage
New
View
Refresh
Export List...
Help
Certificate Template to Issue
Name
Intended Purpose
Administrator
Microsoft Trust List Signing, Encrypting ...
Basic EFS
Encrypting File System
Computer
Client Authentication, Server Authentic...
Directory Email Replication
Directory Service Email Replication
Domain Controller
Client Authentication, Server Authentic...
Server Authentication, SSL Secured Re...
<All>
Encrypting File System, Secure Email, Cl...
Server Authentication
Server Authentication
Enable additional Certificate Templates on this Certification Authority


certsrv - [Certification Authority (LAB-DC01.LAB.PETJE.COM)\lab-LAB-DC01-CA-1\Certificate Templates]
File Action View Help
Certification Authority (LAB-DC0
lab-LAB-DC01-CA-1
Revoked Certificates
Issued Certificates
Pending Requests
Failed Requests
Certificate Templates
Name
Intended Purpose
Administrator
Microsoft Trust List Signing, Encrypting ...
Enable Certificate Templates
Select one Certificate Template to enable on this Certification Authority.
Note: If a certificate template that was recently created does not appear on this list, you may need to wait until
information about this template has been replicated to all domain controllers.
All of the certificate templates in the organization may not be available to your CA.
For more information, see Certificate Template Concepts.
Name
Intended Purpose
OCSP Response Signing
OCSP Signing
RAS and IAS Server
Client Authentication, Server Authentication
RDS
SSL Secured Remote Desktop, Server Authentication
Router (Offline request)
Client Authentication
Smartcard Logon
Client Authentication, Smart Card Logon
Smartcard User
Secure Email, Client Authentication, Smart Card Logon
Trust List Signing
Microsoft Trust List Signing
User Signature Only
Secure Email, Client Authentication
vSphere 6.x
<All>
Workstation Authentication
Client Authentication
OK
Cancel



After following these steps, the created template should be available in the Certificate Templates of the Certificate Authority.

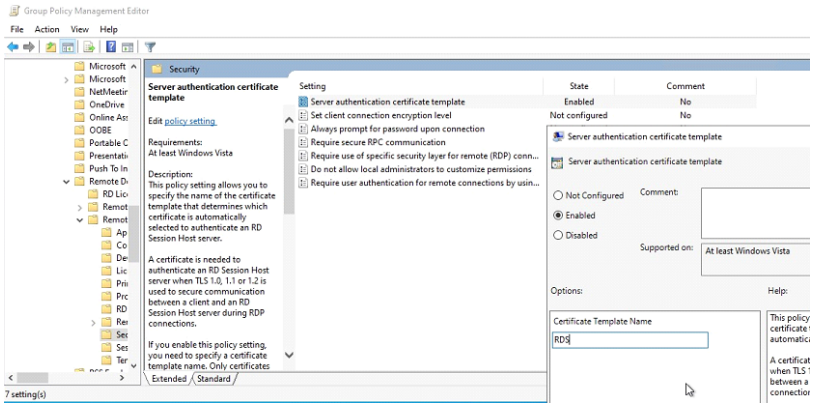
Configure GPO

Final step is to configure a GPO to have the respective machines request the correct certificate and assign that to the RDP listener. The default domain GPO can be adjusted, but a specific GPO with this setting can also be configured.

The settings are:

`Computer Configuration\Policies\Administrative Templates\Windows components\Remote Desktop Services\Remote Desktop Session Host\Security\Server Authentication Certificate Template`

Enter the Certificate Template name as configured in previous steps. In our example `RDS`.

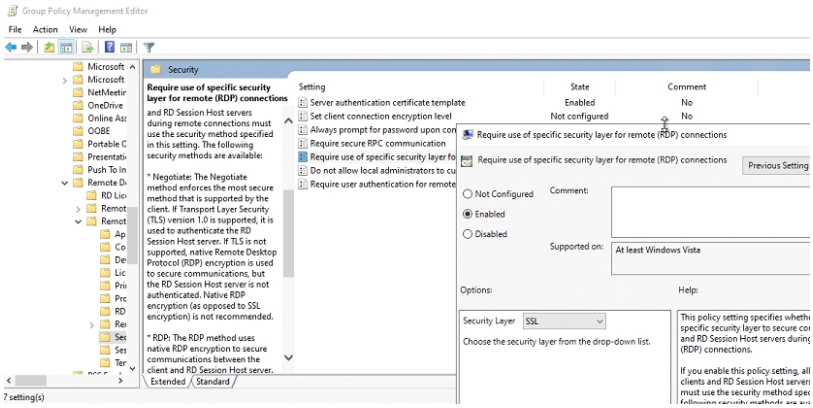


A second setting can optionally be set to force all RDP session to go over SSL. This is however not specifically required for the RDP proxy, but makes RDP connections overall more secure.

The following setting needs to be changed for this:

`Computer Configuration\Policies\Administrative Templates\Windowscomponents\Remote Desktop Services\Remote Desktop Session Host\Security\Require use of specific security layer for remote (RDP) connections`

Change this value to `SSL`.



Next, make sure to have the respective GPO assigned to the target machines in the environment.

After following these steps, Secret Server RDP Proxy should be able to do remote certificate validation against all machines, as long as the issuing CA is trusted by the proxy.

Troubleshooting

In case of errors, make sure to look in the respective proxy logs.

When only web nodes are being used, the proxy information is logged in the Secret Server log files that can be found in: `C:\inetpub\wwwroot\SecretServer\log\ss.log`.

When distributed engines are being used, the relevant log file is on the distributed engine, and can be found in: `C:\Program Files\Thycotic Software Ltd\Distributed Engine\log\ssde.log`.

Error codes can be converted to hex via the following site: https://www.binaryconvert.com/convert_signed_int.html

Enter the decimal error code and perform the conversion. Then search for the resulting HEX value combined with hresult.

Some examples of error codes and explanation can be found below:

1. **HRESULT -2146893019** - Not a trusted authority. The certificate is issued by a certificate authority that is not trusted. Check the certificate to identify the issuing certificate authority and add that in the trusted root certification authorities store on the proxy host.

   ```
   2021-01-14 09:45:00,999 [CID:] [C:] [TID:103] ERROR
   ```

```
Thycotic.RDPProxy.CLI.Session.ProxyConnection - Error encountered in RDP handshake for
client 192.168.178.52:59642 - (null)

System.Exception: HRESULT -2146893019 (SEC_E_UNTRUSTED_ROOT) encountered:

The certificate chain was issued by an authority that is not trusted. at
Thycotic.RDPProxy.SslStream2.ThrowKnownExceptions(Int32 returnCode) at
Thycotic.RDPProxy.SslStream2.CompleteHandshake(String hostname, SspiPacket sspiPacket,
Boolean validateRemoteCertificate)

at Thycotic.RDPProxy.SslStream2.AuthenticateAsClient(String hostname,

Boolean validateRemoteCertificate)

at Thycotic.RDPProxy.CLI.Session.ProxyConnection.

<DoHandshakeAndForward>d__15.MoveNext()
```

2. **HRESULT -2146893022** - Invalid name on cert. Check the machine name you are connecting to and the name on the certificate. In the case of the self signed certificates it is depending if the machine is in a domain or not. When the machine is in the domain the certificate likely has the FQDN, if however the machine is in a workgroup is likely only has the machine name.

```
2021-01-14 10:07:58,045 [CID:] [C:] [TID:15] ERROR

Thycotic.RDPProxy.CLI.Session.ProxyConnection - Error encountered in RDP handshake for
client 192.168.178.117:50535 - (null)

System.Exception: HRESULT -2146893022 (SEC_E_WRONG_PRINCIPAL) encountered:

The target principal name is incorrect.

at Thycotic.RDPProxy.SslStream2.ThrowKnownExceptions(Int32 returnCode)    at
Thycotic.RDPProxy.SslStream2.CompleteHandshake(String hostname, SspiPacket sspiPacket,
Boolean validateRemoteCertificate)

at Thycotic.RDPProxy.SslStream2.AuthenticateAsClient(String hostname, Boolean
validateRemoteCertificate)    at Thycotic.RDPProxy.CLI.Session.ProxyConnection.

<DoHandshakeAndForward>d__15.MoveNext()
```

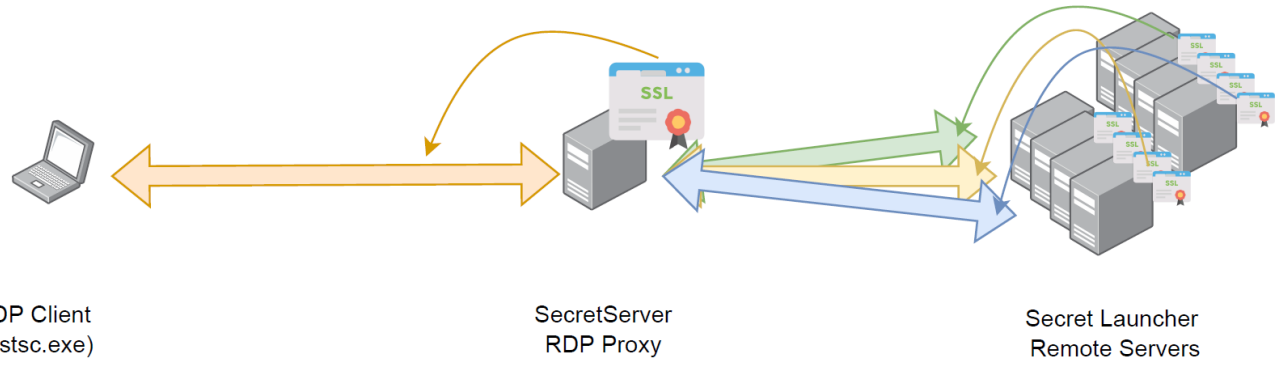
3. **Return code -2146892983** - Incorrect usage for cert. Check the extended usage properties of the certificate that is being used on the target machine. Important is that the certificate has at least the following property: Server Authentication (1.3.6.1.5.5.7.3.1)

```
2021-01-13 15:32:08,086 [CID:] [C:] [TID:104] ERROR

Thycotic.RDPProxy.CLI.Session.ProxyConnection - Error encountered in RDP handshake for
client 192.168.178.105:57804 - (null) System.Exception: Unexpected SSPI handshake return
code -2146892983    at Thycotic.RDPProxy.SslStream2.CompleteHandshake(String hostname,
SspiPacket sspiPacket, Boolean validateRemoteCertificate)

at Thycotic.RDPProxy.SslStream2.AuthenticateAsClient(String hostname, Boolean
validateRemoteCertificate)    at Thycotic.RDPProxy.CLI.Session.ProxyConnection.

<DoHandshakeAndForward>d__15.MoveNext()
```

Disabling Remote Certificate Validation for RDP Proxy

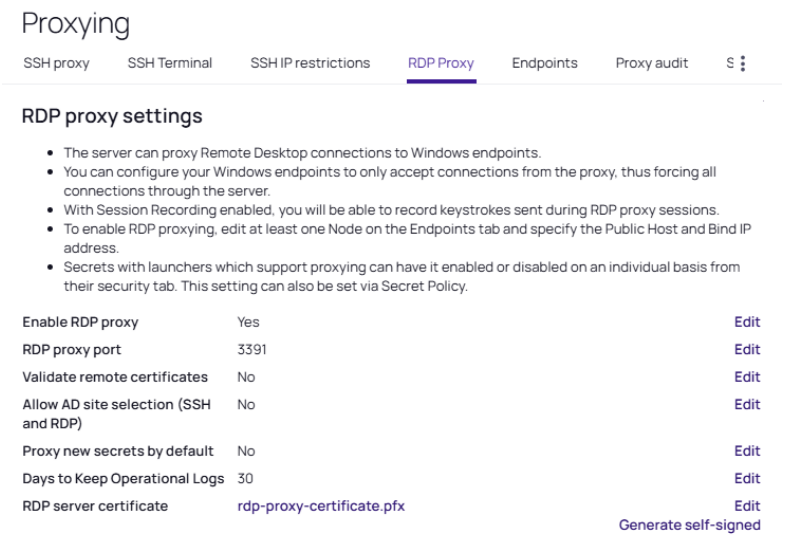
Delinea recommends that you operate in an environment where RDP server certificates are created by a controlled CA and are trusted by machines in the domain. If that is not possible, you can disable remote certificate validation to allow connection to machines that do not serve trusted certificates.

Figure: Normal Remote Certificate Validation for RDP Proxy



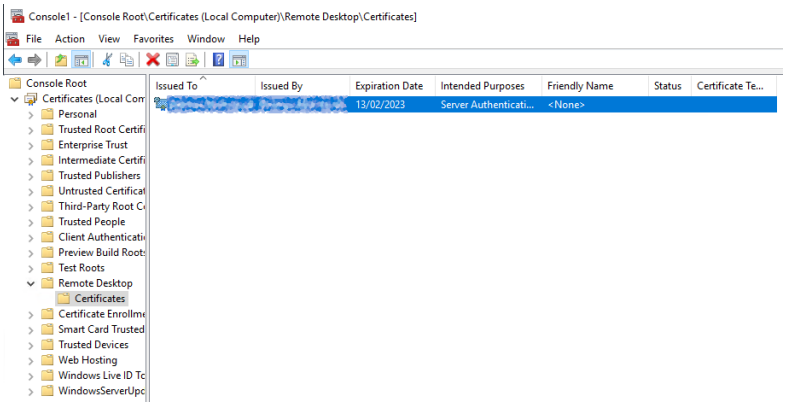
To view or edit your RDP proxy certificate settings:

1. In Secret Server, click the **Settings** button and go to **Proxying > RDP**. The RDP Proxy tab of the Settings page appears:



2. Look at the RDP Proxy setting will have one of two values:
   - Validate Remote Certificates = **No**: Secret Server RDP proxy does not validate the remote server RDP certificate.
   - Validate Remote Certificates = **Yes**: Secret Server RDP proxy validates the remote server SSL Certificate as defined in Local Computer > Remote Desktop > Certificates.

3. To change the certificate settings, in Windows, go to **Console Root\Certificates(Local Computer)\Remote Desktop\Certificates**. The Console appears:



4. Note that these remote server Remote Desktop certificates are self-signed by default.
5. Secret Server RDP proxy cannot validate these unless one of two conditions apply:
   - Each target server certificate is imported in to all the servers hosting RDP proxy.
   - Trusted RDP certificates are deployed.
6. Microsoft describes a process for the second condition in Using certificates in Remote Desktop Services.

SSH and Secret Server

Secret Server provides robust SSH management capabilities, including SSH proxy, command restrictions, terminal administration, and jumpbox routes. The SSH proxy feature routes SSH sessions through Secret Server, ensuring secure and monitored access to endpoints. Command restrictions allow administrators to define and enforce specific commands that users can execute during SSH sessions, enhancing security and compliance. The SSH terminal administration feature enables users to connect to Secret Server via SSH, view and launch secrets, and utilize custom command menus with session recording capabilities. Additionally, SSH jumpbox routes facilitate secure access to internal systems by routing connections through one or more intermediary servers, known as jumpboxes or bastion hosts, which are hardened and monitored to reduce security risks.

SSH Proxy Configuration

The Secret Server proxy routes SSH and RDP sessions and helps protect the endpoint credentials. There are two configuration options for proxying:

- Proxy through the Secret Server Web application
- Proxy through a distributed engine

To learn more about RDP Proxying, please see "RDP Proxy Configuration" on page 817.

Enabling Proxy

1. Go to **Admin > Proxying**.
2. Enable **SSH Proxying**.
3. Generate a new key - click **ECDSA** or **RSA** to generate the related SSH Proxy Host Key in the **SSH Proxy Host Fingerprint**.
4. To enable proxying on Web nodes, under the **Endpoints** tab, in the **Nodes** section, edit the row to set the **Public Host** and **Bind IP Address**. For a standard server, these can be the same, but if the public IP of the server is not set on the server (such as a load balancer or an EC2 instance with an elastic IP), they will be different. Enter the **Available Port Range** if prompted.
5. To enable proxying for a specific site and all engines within that site, edit the row in the **Sites:** section. In the Update Proxy Configuration For Site popup specify:
   - **Enable SSH Proxy:** Check to enable SSH Proxy.
   - **SSH Proxy Port:** Specify the related SSH Proxy Port if prompted.
   - **Inherited (Port 22):** Check to enable the defaulted Port 22.
   - **Available Port Range:** Specify the range of the available ports.
   - **Enable RDP Proxy:** Check to enable RDP Proxy. See RDP Proxy Configuration for more details.
   - **RDP Proxy Port:** Specify the related RDP Proxy port if prompted.
   - **Inherited (Port 3390):** Check to enable the defaulted Port 3390.
6. The engines for the sites are listed in the **Engines** section below. The **Public Hostname or IP Address** is the public host or IP the launcher connects to and the **Bind IP Address** is the IP on the server that the SSH proxy is listen on. Again, these will typically be the same, but may be different if the resolvable IP or host of the engine machine is different than the IP on the network adapter on the machine.
7. Enable proxying on a secret with a PuTTY launcher. The launcher now connects to the assigned site, which is set on the **General** tab. If the site has proxying enabled, it will go through the engines available in the site, otherwise it will use the Secret Server Web application proxy.

Web Application Proxy Performance

Minimum Hardware

- Intel 3.7 GHz Quad Core
- 16 GB of RAM
- 100 MB/s plus network capability

Session Activity

We tested sessions with standard usage, such as opening and modifying files and navigating the file system on Linux. On Windows, the activity was opening MMC snap-ins, editing files, and copying files through the RDP session. If you have constant large file transfers across multiple concurrent sessions or otherwise transferring large amounts of data (such as streaming a video through an RDP session), the maximum concurrent sessions will be significantly reduced.

Table: Concurrent Proxy Sessions

Protocol	Concurrent Sessions
SSH	300
RDP	100

Note that the file transfer speeds when using SSH Proxy with tools such as FileZilla or WinSCP in Secret Server may be slower than direct, non-proxied transfers. Recent improvements have enhanced transfer speeds compared to previous performance; however, some degradation is still expected due to the nature of the proxying process.

For optimal performance:

- Ensure your network connection has low latency and sufficient bandwidth.
- Consider alternatives for large file transfers, such as direct SCP commands or other non-proxied methods, if proxying is not essential for your use case.

This behavior is inherent to the security and architecture of the SSH Proxy feature and is not indicative of a system malfunction.

Proxy Connections

Connections from the user to the proxy are over SSH, and you can configure the port. The user's machine will connect to either an engine SSH proxy or the Secret Server Web application SSH proxy.

Figure: Default Secret Server Web Application Proxy (example)

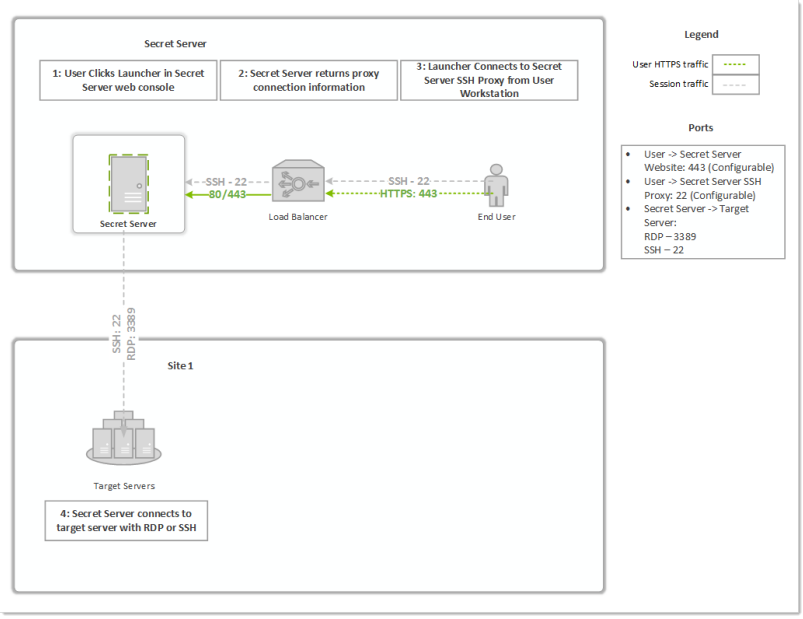
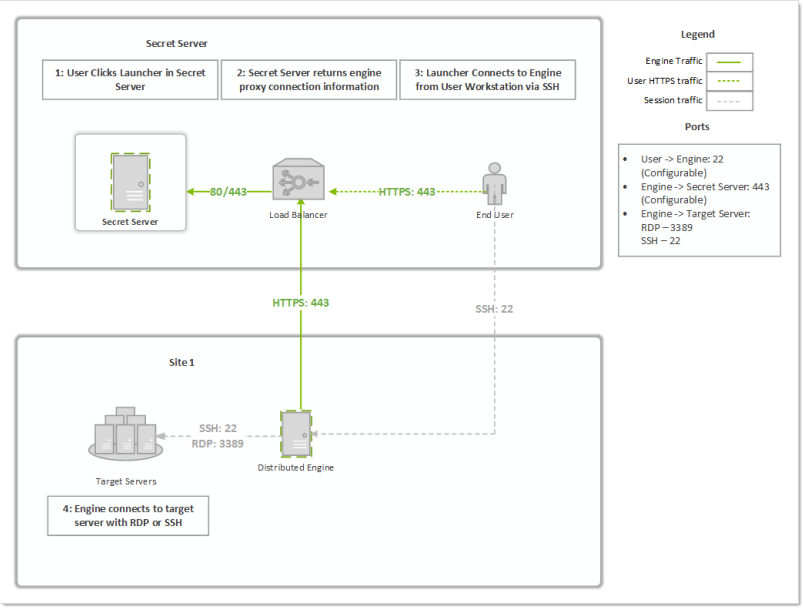


Figure: Proxy through a Distributed Engine (example)



SSH Proxy with Multiple Nodes

If you are using clustering with Secret Server, you can pick exactly which of your nodes act as a SSH proxy by going to the **Admin > Proxying** page and scrolling down to the **Nodes** section. For each node you wish to be a proxy, configure the **SSH Public Host** (must be an IP address, not a DNS name) and the **SSH Bind IP Address** (use 0.0.0.0 to easily bind to all IPv4 Ps on a server). There is no need to configure all nodes if you do not want them all to be proxies.

As soon as the IPs are saved for each node, the node should start listening on the SSH proxy port. You can verify that with netstat. If you do not see the node listening on your chosen port, perform an IIS reset and hit its Secret Server website. It should be listening once Secret Server starts up again. For example:

```
C:\Users\Administrator>netstat -ano | find ":22"
TCP  0.0.0.0:22  0.0.0.0:0  LISTENING  3600
```

Now, when a user connects to the Secret Server Web page, if the node they are hitting is setup to be a SSH proxy, they will connect to that node's SSH public host IP. If the node they are connected to is not setup to be a SSH proxy, then users will round robin between the other nodes that are SSH proxies and connect to their SSH public host IP.

SSH Terminal Administration

Introduction

This document discusses using an SSH terminal with Delinea Secret Server.

Feature Summary

- Connect using SSH to a terminal hostname and port to log in to terminal and run commands
- Display custom terminal banner after successful connection
- Display available commands on successful login (display again with `man` command)
- Log in to the terminal as a Secret Server user (SSH Proxy must be enabled)
- Can set an inactivity timeout. Can be set to *disabled* or with a two-minute minimum.
- Start a terminal connection and launch in a single line. For example:
  `ssh <user>@<ss_ip> -t launch <secret_id>`
- Use two-factor authentication (2FA) for access (optional)
- Use the SSH terminal interface to Secret Server for viewing and launching secrets
- Use these commands:
  - `Man` command to display detailed command description
  - `Search` command to display matching secrets
  - `Cat` command to display secret details of with specified secret ID
  - `Launch` command to begin a Proxy launch session with specified secret ID
- Use up and down keystrokes for command history
- Supports custom SSH command menus and session recording logging

Requirements

System Requirements

- Secret Server 10.7.000000 Secret Server
- Secret Server**Professional** or **Platinum** Edition license

Recommended

"Installing RabbitMQ" on page 93

Secret Server Permission Requirements

Admin:

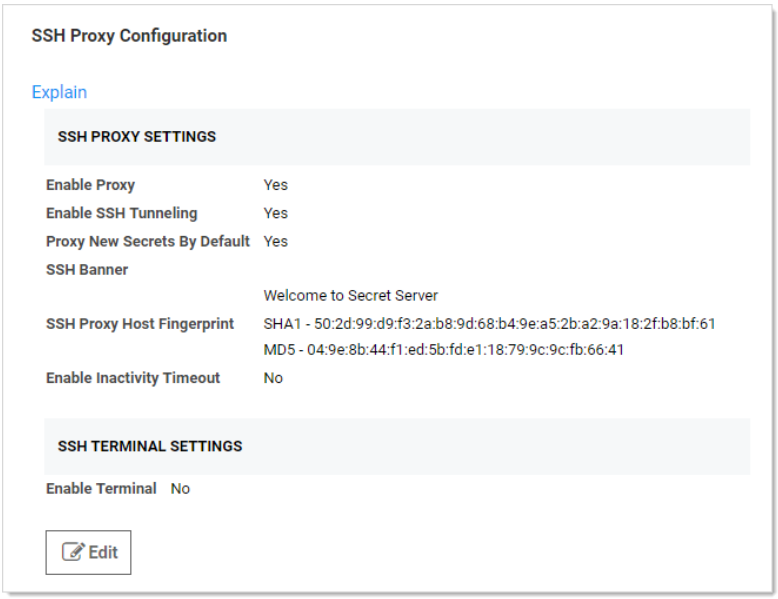
- Administer Configuration
- Administer Proxying Configuration
- View Configuration
- View Proxying Configuration

User: View Secret

Configuring SSH Terminal

Enabling SSH Terminal on Secret Server

1. Prerequisites:
   - Must meet Admin permission requirements (see Secret Server Permission Requirements)
   - Secret Server**Professional** or **Platinum** Edition license
2. Navigate to **Secret Server > Admin > Proxying**.

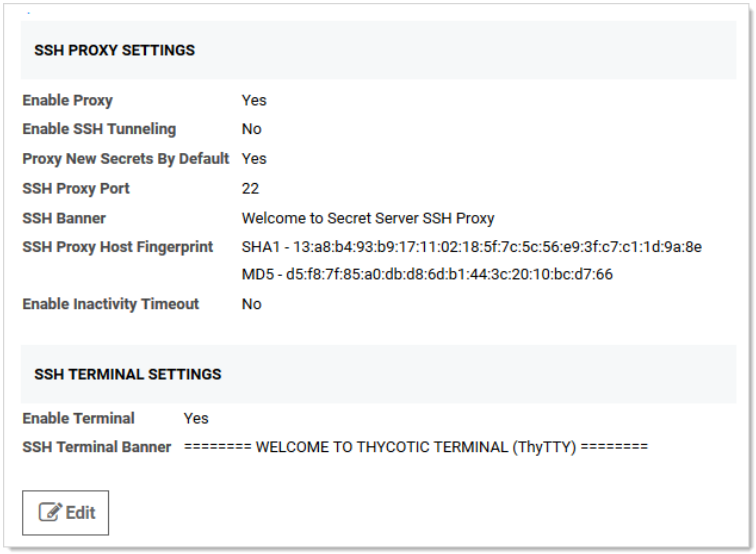


3. Click the **Edit** button.
4. Type your SSH proxy configuration settings (see "Configuring SSH Proxies for Launchers" on page 661):
   a. Enable **SSH Proxy** (required to use SSH terminal).
   b. (optional) Enable **Proxy New Secrets by Default**.

      To launch a secret via the terminal, the secret must have proxy enabled. Only SSH-based credentials can be launched in the terminal.

   c. Click to enable **SSH Terminal**.
   d. (optional) Customize the **Terminal banner** for your environment.
   e. (optional) Click to enable **Terminal Inactivity Timeout** (in seconds).

f. The resulting settings should look something like this:



5. Specify the IP address for nodes (and engines) that will run SSH proxy:
   a. Navigate to **Admin > Proxying > Nodes.**
   b. Set the **SSH Public Host.** This is the public hostname or IP that the client launcher connects to. In most cases, this is the same as the SSH bind address; however, there are cases where the public IP or host differs from the private IP that Secret Server should bind to, such as NAT or Amazon EC2 instances.
   c. Set the **SSH Bind IP Address**. This defaults to (0.0.0.0). The IP Address of the network adapter that the Secret Server SSH listener should bind to. This should not be localhost or 127.0.0.1. If you are not sure which bind IP address to use, you may use 0.0.0.0, which binds to all IPv4 interfaces on the machine.

Enabling Terminal on Secret Server Distributed Engine

SSH terminal can also run on each proxy-enabled distributed engine (DE) site.

> To launch secrets on non-local sites, users **must** connect to an SSH terminal over an engine on this site.

1. Go to **Admin > Proxying > Sites**.
2. Click to select **Proxy Enabled**.
3. Type an **SSH Port**.
4. Go to **Admin > SSH Proxy > Engines**.
5. Type the **Hostname** and **IP Address** (description above).
6. Type the **SSH Bind Address** (description above).

Logging into the SSH Terminal

1. From any SSH terminal, connect to hostname or IP address and port, as specified in the SSH Proxy Configuration page. Use the DE hostname or IP if connecting to an engine. Examples:

   ```
   ssh 127.0.0.1 -p 22
   ssh user54@127.0.0.1 -p 22
   ```

2. If not provided in the SSH connect command, enter your Secret Server username and password at the **Login as:** prompt.
3. If successful, you will see the terminal banner displayed, along with a list of available commands.

Increasing Maximum Concurrent Logins for Users

Logging in to SSH terminal counts against the number of concurrent Secret Server sessions a user is allowed. For example, if **Maximum concurrent logins per user** is set to "1" and the user john.smith is logged into the Secret Server Web user interface, then john.smith logs into SSH terminal, his first Web session will end, and he will have to log in again to use the Web user interface.

To increase the maximum concurrent logins per user:

1. Go to **Admin > Configuration**. The Configuration page appears.
2. Click the **Login** tab.
3. Click the **Edit** button at the bottom of the page. The page becomes editable.
4. Click the **Maximum concurrent logins per user** dropdown list and select the desired number.
5. Click the **Save** button.

SSH Terminal Login with Two Factor Authentication

SSH terminal is considered a Web service and can be used with two factor authentication (2FA). To enable 2FA for terminal:

1. Follow the steps under "Multi-Factor Authentication" on page 433 to set up 2FA.
2. Go to **Admin > Configuration > Login > Require Two Factor for these Login Types** and select one of these:
   - **Website and Web Service Login**
   - **Web Service Log on Only**
3. Enable 2FA on the Secret Server user by going to **Admin > Users > Select a user > Edit > Two Factor** and select the 2FA option.

   > FIDO2 authentication is not supported in this version of SSH terminal.

4. From any SSH terminal, connect to hostname or IP address and port, as specified in the SSH Proxy Configuration page. Use the distributed engine hostname or IP if connecting to an engine. Examples:

   ```
   ssh 127.0.0.1 -p 22
   ssh username@127.0.0.1 -p 22
   ```

5. If not provided in the SSH connect command, enter your Secret Server username and password at the **Login as:** prompt.
6. You will be prompted for a PIN or custom challenge message by your 2FA provider. Example:

   login as: `duouser`**<Enter>**

   Using keyboard-interactive authentication:

   duouser@127.0.0.1's password: `uewori#$%tdtd`**<Enter>**

   Using keyboard-interactive authentication:

   Pin Code: `3787`**<Enter>**
7. If successful, you will see the terminal banner displayed, along with a list of available commands.

Escaping Special Characters

When manipulating secrets containing special characters, such as single quotes and double quotes, you must escape those characters in the command.

Example: To search for an item with a space in the name, put the name in single or double quotes:

`search "My Secret"` or `search 'My Secret'`

Example: To search for an item with a single quote embedded in the name, there are two options:

- Encase the term in double quotes:
  `search "Bob's Secret"`
- Escape the single quote with a backslash:
  `search 'Bob\'s Secret'`

Example: Similarly, to search for an item with a double quote embedded in the name, there are two options:

- Encase the term in single quotes:
  `search '"Weird" Secret'`
- Escape the internal double quotes with a backslash:
  `search "\"Weird\" Secret"`

Terminal Commands

man

Syntax

```
man [command name]
```

Description

Displays command help for specific or all commands. *Man* is short for *manual*.

Examples

```
man
```

Short help for all commands.

```
man cat
```

A detailed description of the `cat` command.

search

Syntax

```
search [-st] <search_text> [-f <folder_id>] [-fav] [-r] [-sf <search_field>] [-skip <skip_
results>] [-s] [-t <secret_template_id>] [-take <max_results>]
```

Description

Returns a list of Secret Server secrets by keyword, which you can filter using several command-line switches.

Parameters

`-st <search_text>`

Required. Text to search for. `-st` is optional. Returns 25 results by default. Use `-take` to change from the default.

`-f <folder_id>`

ID of the secret folder to limit the search to.

`-fav`

Only search "favorite" secrets.

`-r`

Ignore restricted secrets in the search. Restricted secrets are included by default.

`-s`

Ignore subfolders in the search. Subfolders are included by default.

`-sf <search_field>`

ID of the secret field to limit the search to. Potential fields, which vary by secret template, can include the following examples:

- Address1
- Address2
- Address3
- Blog
- CardType
- City
- Combination
- Contact Number
- Country
- Email Address

- ExpirationDate
- Fax
- First Name
- FullName
- Home Phone
- Last Name
- Machine
- Mobile Phone
- Notes
- Number
- Password
- Pin
- PinCode
- Server
- SSN
- State
- Username
- Website
- Work Phone
- Zip

These fields match those on the REST API endpoint.

`-skip <skip_results>`

Skip this number of initial results. Useful for processing "pages" of results.

`-t <secret_template_id>`

Only search secrets based on the template with this template ID.

`-take <max_results>`

Take a total of only this number of results. Useful for processing "pages" of results. Defaults to 25 results.

Examples

```
search -st admin
```

Find a list of secrets matching "admin." Returns 25 results (the default).

```
search admin
```

Same search using alternate syntax. `-st` is not required.

```
search -st jones -fav
```

Find a list of "favorite" secrets matching "jones" in any field Returns 25 results (the default).

```
search admin -take 50
```

Outputs a list of secrets matching "admin", up to 50 results.

```
search Zardoz -take 50 - skip 50 -sf "Secret Name"
```

Find a list of secrets with "Zardoz" in the "Secret Name" field. Return 50 results, starting with the 51st secret found.

```
search admin -skip 25 -r
```

Find a list of secrets matching "admin" in any field. Return 25 results, which is the default. Skip the first 25 results. Ignore restricted secrets.

cat

Syntax

```
cat [-s|-id|-secret-id] <secret_id> [-c|-comment <comment_or_access_request>] [-t|-ticket
<ticket_number>] [-ticketsystemid <ticket_system_id>]
```

Description

- Displays information on a secret. The available information depends on the secret's template. *cat* is short for *concatenate*.
- Catches access errors, such as "comment required" or "requires approval", and displays them on the terminal
- Audits "view" comments.
- Provides launch connection command instructions. Shows the correct launch parameter and a connection string (if the terminal connection and the site on the secret do not match).

> If a required access element is not provided in the command, the terminal will respond with an error that should indicate what is missing.

Parameters

```
[-s|-id|-secret-id] <secret_id>
```

Required. The secret ID. Three optional switches.

```
[-c|-comment <comment_or_access_request>]
```

The text for the comment or access request.

```
[-t|-ticket <ticket_number>]
```

The ticket number for the request.

```
[-ticketsystemid <ticket_system_id>]
```

The unique ticket system ID.

Examples

```
cat 24
```

Display the contents of the secret with the ID 24. Only works after access is approved.

```
cat -id 24
```

Alternate syntax. Display the contents of the secret with the ID 24.

```
cat -id 25 -comment "Viewing this secret"
```

Add a "view" comment to, and then display the contents of the secret with the ID 25.

```
cat -id 26 -comment "Requesting view access to install software" -ticket 123 -ticketsystemid
2
```

Add an "access request" comment to the secret with the ID 26. Assign the request the ticket number 123 and the ticket system ID of 2 to that request.

> The most common secret restrictions are "requires view comment" or "requires access request." The `-comment` parameter takes care of both of these because the underlying API call (`SecretAccessCreateArgs`) is agnostic.

launch

Syntax

```
launch [-s|-id|-secret-id] <secret_id> [-m|-Machine <machine_name>] [-c|-comment <view_
comment_or_approval_request_reason>] [-t|-ticket <ticket_number>] [-ticketsystemid <ticket_
system_id>]
```

Description

- Creates a proxy connection to the machine
- Secret must have proxy enabled
- Supports launch from secrets with private keys
- Audits launches

Parameters

```
[-s|-id|-secret-id] <secret_id>
```

Required. The secret ID. Three optional switches.

```
[-c|-comment <comment_or_access_request>]
```

The text for the comment or approval request.

```
[-m|-Machine <machine_name>]
```

Machine name for the launch. This may be required if a customized secret template does not contain a machine field or a launcher requires a machine entry on launch.

```
[-t|-ticket <ticket_number>]
```

The ticket number for the request.

```
[-ticketsystemid <ticket_system_id>]
```

The unique ticket system ID.

Examples

```
launch 24
```

Begins the SSH proxy session with the secret with the ID 24 and the specified credentials and machine. Only works after access is approved.

```
launch -id 24
```

Alternate syntax. Begins the SSH proxy session with the secret with the ID 24 and the specified credentials and machine. Only works after access is approved.

```
launch -id 25 -comment "Launching this secret"
```

Submits a "view" comment to the secret with ID 25. Begins the SSH proxy session with secret credentials and machine.

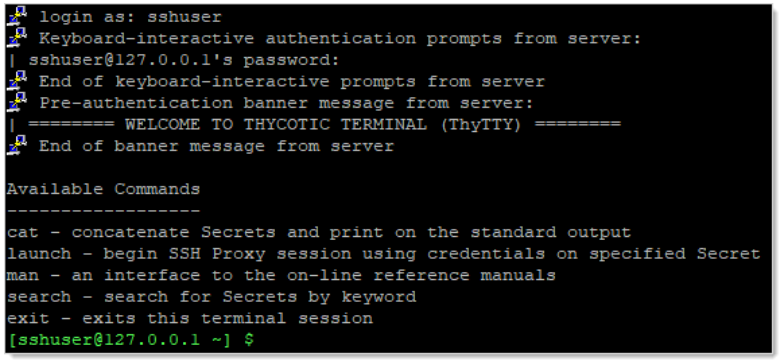
```
launch -id 26 -machine XYZ -comment "Requesting view to launch temporary sudo account for the XYZ machine"
```

Submits an "access request" comment to the secret with ID 26 on the machine XYZ with the ticket number 123 and ticket system ID 2.

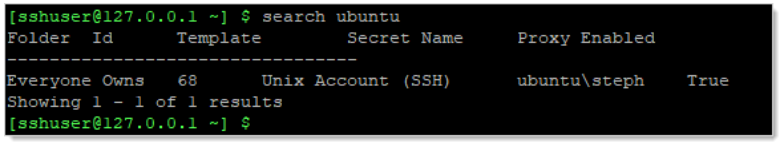
Launching a Secret with the SSH Terminal

Launching a Secret on a Local Site

1. To launch, the secret must be:
   - Enabled for proxy **(Secret Server > Secret > Security > Enable Proxy)**
   - Shared with the terminal user
2. Log in to the terminal with Secret Server user credentials:



3. If the secret ID is unknown, search for the desired secret with the search command:



4. To view secret detail, get the secret ID from search results, and run

   ```
   cat <secret_id>
   ```

```
[sshuser@127.0.0.1 ~] $ cat 68
Secret Name:     ubuntu\steph
Secret ID:       68
Secret Type:     Unix Account (SSH)
Folder:          \Everyone Owns
Launch Enabled: True
Machine:         [redacted].thycotic.com
Username:        steph
Password:        **********

Can be launched with command:

launch 68

[sshuser@127.0.0.1 ~] $
```

5. To launch the secret, enter the launch command as specified in the last line of secret details:
   `launch <secret_id>`

```
[sshuser@127.0.0.1 ~] $ launch 68
Secret Server Launch: Secret ID 68 found. Attempting launch...
Connected to Target! Attempting Authentication...
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-151-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Jul 24 13:07:53 EDT 2019

  System load:  0.0               Processes:              101
  Usage of /:   40.2% of 14.58GB  Users logged in:        0
  Memory usage: 22%               IP address for eth0:    192.168.60.252
  Swap usage:   0%                IP address for docker0: 172.17.0.1

  Graph this data and manage this system at:
    https://landscape.canonical.com/

82 packages can be updated.
63 updates are security updates.

Last login: Wed Jul 24 12:59:59 2019 from 192.168.68.137
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

steph@ubuntu:~$
```

6. To exit the launch session and return to the terminal, type `exit`.

```
Last login: Wed Jul 24 12:59:59 2019 from 192.168.68.137
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

steph@ubuntu:~$ exit
logout
Socket was shutdown.
[sshuser@127.0.0.1 ~] $
```

7. To exit the terminal session, type `exit` again.

Launching a Secret on a Distributed Engine Site

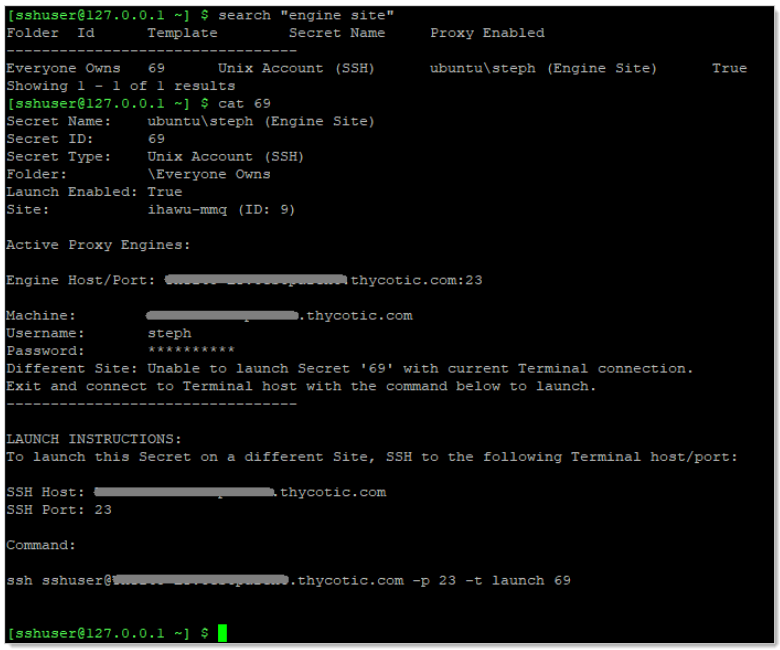
1. To launch, the secret must be:
   - Enabled for proxy **(Secret Server > Secret > Security > Enable Proxy)**
   - Shared with the terminal user
2. Log in to the terminal with Secret Server user credentials:

```
login as: sshuser
Keyboard-interactive authentication prompts from server:
| sshuser@127.0.0.1's password:
End of keyboard-interactive prompts from server
Pre-authentication banner message from server:
| ======== WELCOME TO THYCOTIC TERMINAL (ThyTTY) ========
End of banner message from server

Available Commands
------------------
cat - concatenate Secrets and print on the standard output
launch - begin SSH Proxy session using credentials on specified Secret
man - an interface to the on-line reference manuals
search - search for Secrets by keyword
exit - exits this terminal session
[sshuser@127.0.0.1 ~] $
```

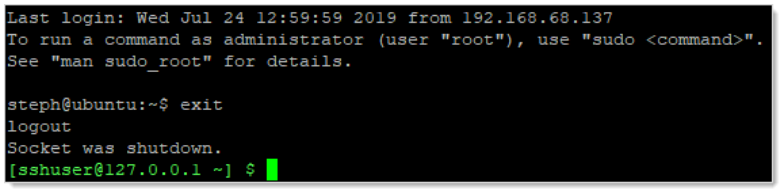
3. If secret ID is unknown, search for the desired secret with the search command:

```
[sshuser@127.0.0.1 ~] $ search ubuntu
Folder  Id      Template        Secret Name     Proxy Enabled
---------------------------------
Everyone Owns   68      Unix Account (SSH)      ubuntu\steph    True
Showing 1 - 1 of 1 results
[sshuser@127.0.0.1 ~] $
```

4. To view secret detail, get the secret ID from search results, and run `cat <secret_id>`



5. Note that the connection is not made, and instructions are displayed for logging into another distributed engine terminal to launch the secret.
6. Note the suggested parameters in the **Launch Instructions**.
7. Type `exit` and press **<Enter>** to disconnect from the current session:



8. Open a new SSH session suggested parameters:

   `ssh <secret_server_username>@<engine_hostname_or_ip> -p <Port> -t launch <secret_id>`
9. Enter the password to log in, and the secret should immediately launch.

Launching a Secret upon Terminal Connection

1. To launch, the secret must be:
   - Enabled for proxy **(Secret Server > Secret > Security > Enable Proxy)**
   - Shared with the terminal user
2. If the secret ID and connection string is known, you can log in and immediately launch the secret with the following command:

   `ssh <secret_server_username>@<hostname_or_ip> -p <Port> -t launch <secre_id>`
3. If you do not know that connection string, log into terminal and run:

```
cat <secret_id>
```

4. Look at the **Launch Instructions** at the end of secret details, and note the parameters.

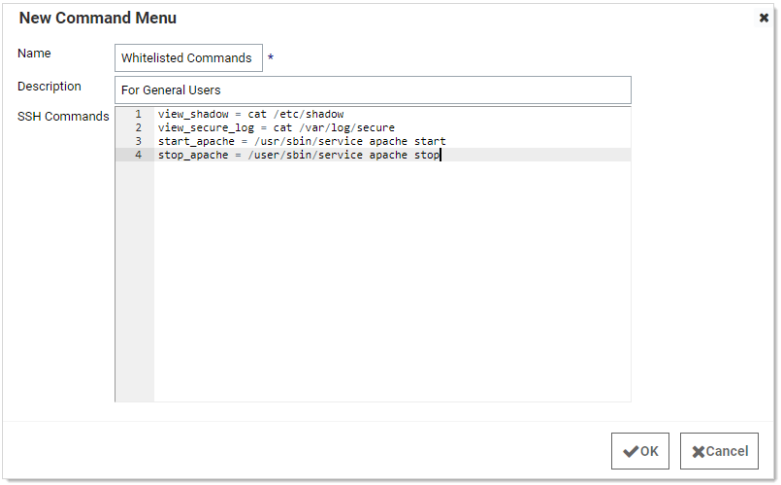
> Note that if a secret is not checked-out, it will be silently checked-out when launching via SSH Terminal. The secret will be silently checked-in when the SSH session is terminated. If the secret is already checked-out by a different user, Terminal will display that it is checked-out.

SSH Terminal Launching with a Custom SSH Command Allowlist

Secret Server terminal can launch secrets with custom SSH Command restrictions. For detailed instructions on SSH command menus, please see"Managing Superuser Privilege" on page 678.

> Custom SSH command menus require either the Secret Server Platinum or Unix SUPM add on license.

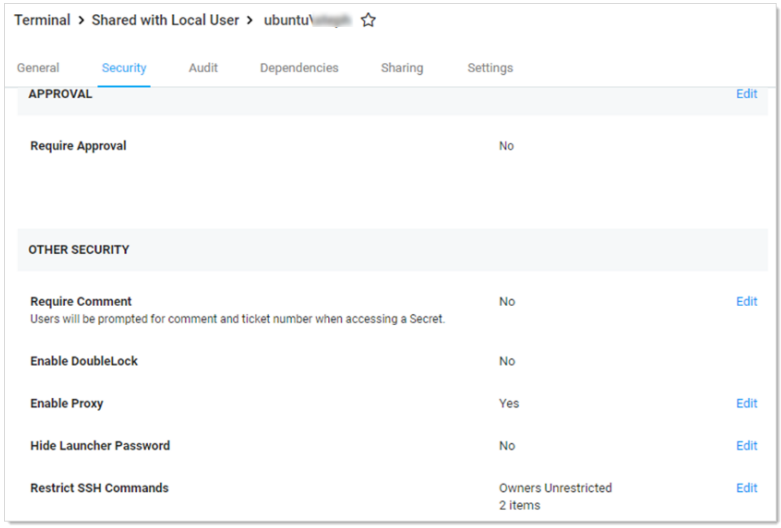
1. Go to **Admin > See All**. The Administration page appears.
2. Click the **SSH Command Menus** link. The SSH Command Menus page appears.
3. Click the **Create New** button.
4. Type a name, description and the SSH commands:



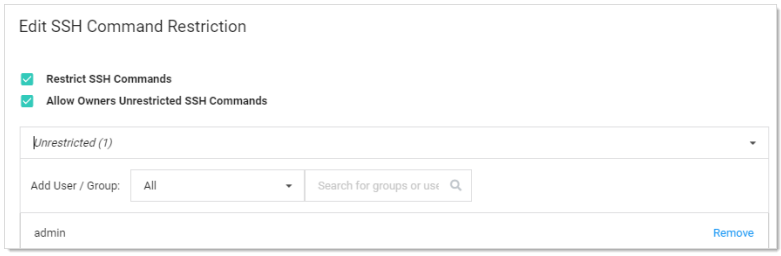
Once one or more command menus have been created, access can be controlled to individual Unix SSH secrets.

5. On the **Security** tab of a secret that can use a proxied SSH session, proxy must be enabled, as well as command menu restrictions. If **Allow Owners Unrestricted SSH Commands** is enabled, any user who is an owner of the secret has unrestricted use of the launched session. That is, that user is able to type in commands as in a normal SSH session. Additionally, other groups can be assigned the unrestricted role as well.
6. In the following example, the "admin" group is unrestricted, and everyone who is not in that admin group is restricted to only being able to run the allowlisted commands that are specified in the user command menu

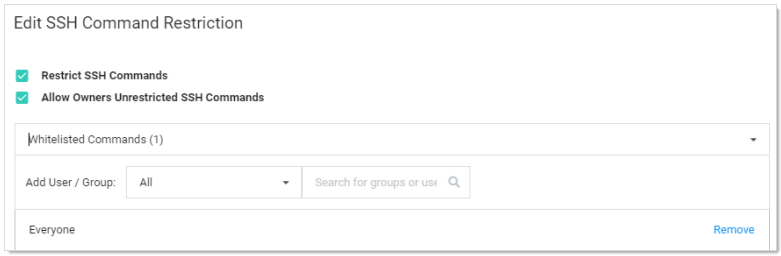
created above.



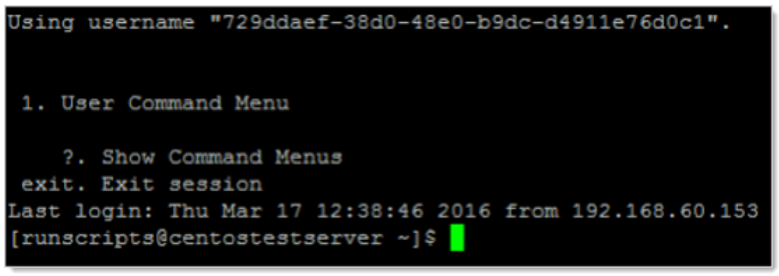
When you click the Edit link:



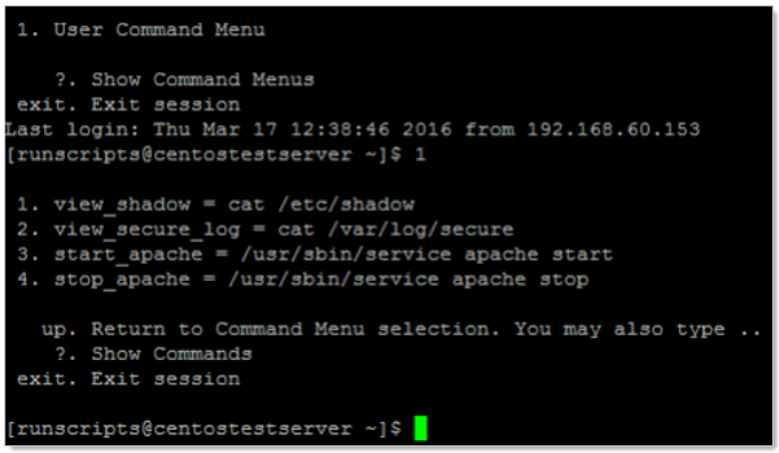
And click the dropdown list to select Allowlisted Commands:



7. A user who is subject to SSH command restrictions is presented with a screen similar to the following when launching this secret from Secret Server terminal:



The user simply enters the number of the command menu to see available commands or types "?" to display the options again:



Only the commands listed can be run by this user. The user can either enter the number of the command to be run, or the name of the command, which is the word to the left of the equal = sign. Other options are available (as shown) to navigate through the available command menus, display help, or exit the session.

SSH Terminal Launching with Session Recording

Secret Server SSH terminal launches also support session recording for session client or server data. When a user launches a secret with session recording enabled through SSH terminal, session data is available in the Secret Audit tab as session data.

> Session recording requires either Secret Server Platinum or the session recording add-on license.

> See the Session Recording section for more information.

To enable session recording:

1. Go to **Admin > Configuration**. The Configuration page appears.
2. Click the **Session Recording** tab.
3. Click the **Edit** button at the bottom of the page. The page becomes editable.
4. Ensure the **Enable Session Recording** check box is selected.

5. Modify other settings as desired.
6. Click the **Save** button.

To enable session recording on a secret:

1. Open a secret.
2. Click the **Security** tab.
3. Click the **Edit** link to the right of the **Session Recording Enabled** setting. The Edit Security popup page appears.
4. Click to select the **Session Recording Enabled** check box.
5. Click the **Save** button.

To view session data following a terminal secret launch:

1. Open a secret.
2. Click the **Audit** tab.
3. Find the **LAUNCH** action in the table.
4. Click the **View SSH Session Log** link.

SSH Key Pairs for Terminal

Overview

SSH key pairs allow users to authenticate to Secret Server terminal without using a password. The user generates a key pair in Secret Server, at which time the private key can be downloaded by the user locally in the format they require. The key pair generation process is the only time the private key will be provided to the user. If this private key is lost, the user must log back into Secret Server and generate a new public/private key pair.

Limitations

- Currently users can only authenticate to Secret Server using SSH keys by using Secret Server's SSH terminal.
- Only PuTTY and OpenSSH keys can be generated.

Enabling Users to use SSH Key Pairs to Authenticate

There are three requirements for enabling Public SSH Keys:

- SSH Proxy is enabled in Secret Server.
- SSH Terminal is enabled in SS.
- SSH key integration is enabled in SS's Configuration > Login settings. To do so:
  1. **Unix Authentication Method**: choose **Public Key only**, **Password or Public Key** or **Password and Public Key** to enable SSH key pair authentication.
  2. Once done, the admin can also set an optional expiration time frame for the public SSH keys, which applies to all users.

Once these 3 requirements have been met, users can use the main navigation to create SSH key pairs.

Creating SSH Key Pairs

An SSH key pair consists of a private key and a public key. Only the public key is stored in the user's settings—the private key downloaded during generation is **not** saved inside Secret Server and should only be available to the user, to remain secure.

During terminal login, if the user provides a private key for authentication, Secret Server validates the provided private key against the user's available (and enabled) saved public keys. If a key pair match is found, the authentication succeeds (or the next required authentication step, for example a password prompt, is shown).

For security reasons, only users can create their own SSH key pairs. However, Secret Server Administrators can deactivate any user's public SSH keys as follows:

1. Navigate to the **Public SSH Keys** page using the main navigation at the top right of the page.
2. Click the **Create SSH Key** button above the grid, then fill out the form in the popup page.
3. Click the **Create SSH Key** button in the popup. After a moment you will be able to save the private key.

Administering Public SSH Keys

1. Navigate to the User by going to **Admin > Users**.
2. Locate the user in the dropdown list and select it.
3. On the **General** tab click the **Administer Public SSH Keys** button. You can now deactivate the user's public SSH keys.

Using SSH Keys for Authentication (PuTTY Example)

1. In PuTTY, fill in the **Session** view to match your SSH proxy connection settings in Secret Server.
2. In the **SSH > Auth** section of PuTTY, add the private key file that was saved when generating the key in Secret Server.
3. You will be prompted to enter your passphrase for the key if one was set.
4. You will be prompted to enter your password if **Unix Authentication Method** also requires a password.

SSH Command Restrictions

SSH command restrictions in Secret Server enhance security by allowing administrators to define and enforce specific commands that users can execute during SSH sessions. This feature, part of the privilege management capabilities, requires SSH proxy to be enabled and supports the creation of command menus that map user-friendly command names to system commands. Administrators can configure these menus to restrict users to a predefined set of commands, preventing unauthorized or potentially harmful actions. Command restrictions can be applied to individual secrets or through secret policies, ensuring consistent enforcement across the organization. This setup not only minimizes the risk of misuse and accidental errors but also supports compliance with security policies by providing detailed audit logs of all executed commands.

SSH Blocked Command Lists

Overview

Secret Server supports privilege management and command restrictions for UNIX and other platforms with SSH interfaces. Privilege management is an additional layer of access control that you can apply to secrets with SSH launchers over SSH proxy. With privilege management, you can grant users access to a machine to block specific commands that a user may run as root or any other privileged account.

To use command restrictions, Secret Server must have SSH Proxy and Enable Block Listing enabled.

With SSH blocked command lists, you can define disallowed commands when connecting as a privileged account. The blocked command list is defined by a series of regular expressions.

Upon launching a secret with an assigned SSH command blocklist, each command sent to the target is evaluated for a match on the blocklist. If the command is found to match a list entry, that command is blocked from execution. Blocked output is shown to the user at the terminal.

Requirements

System requirements:

- Secret Server 10.11 or later
- Secret Server Platinum Edition license or Secret Server Professional and Unix SUPM license
- SSH proxy must be enabled

Creating SSH Blocked Command Lists

The format for specifying a blocked command follows a regular expression syntax that is typical to most scripting languages. The blocked commands are surrounded with `\b` word boundary anchors. For example, to block the `sudo` command, you use:

```
\bsudo\b
```

This expression blocks the execution of any command with the `sudo` string in it (as a separate word), for example, these are blocked:

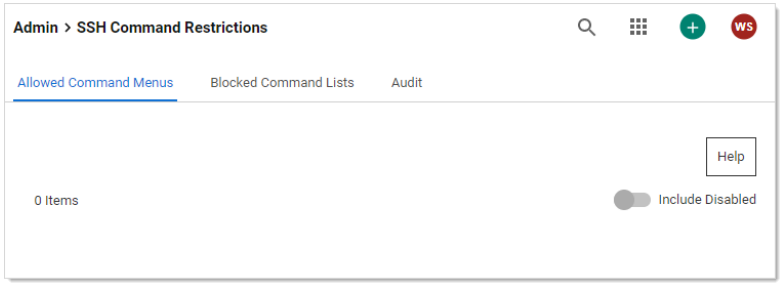
- `sudo`
- `sudo root`
- `sudo ls /usr/local/protected`
- `sudo shutdown -r +15 "quick reboot"`

And these are not:

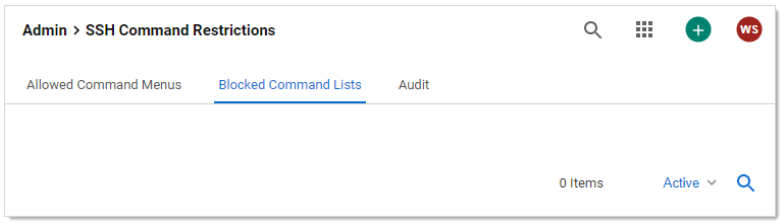
- `cat sudoku`
- `echo "sudo"`

To create a list of blocked commands:

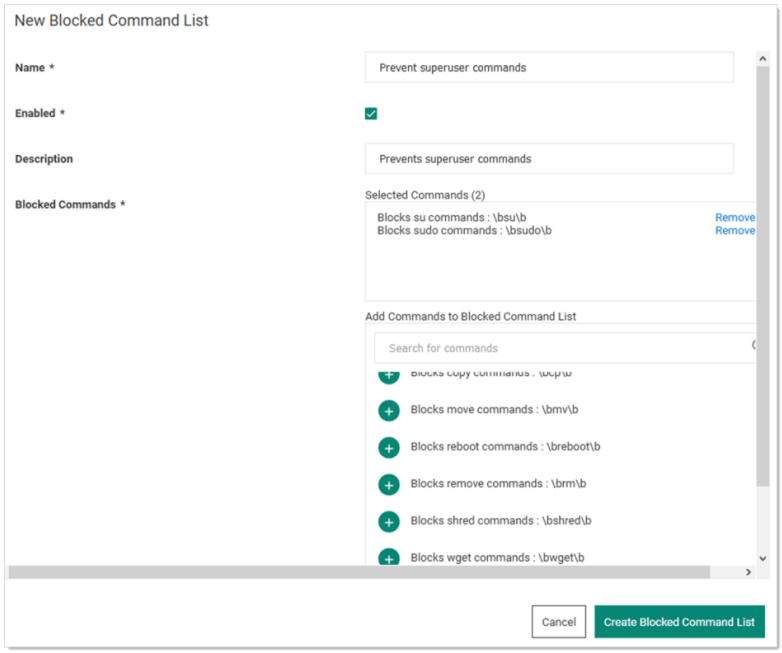
1. Go to **Admin > See All**.
2. Hover the mouse pointer over the **Actions** menu item and select **SSH Command Restrictions**. The SSH Command Restrictions page appears:



3. Click the **Blocked Command List** tab:

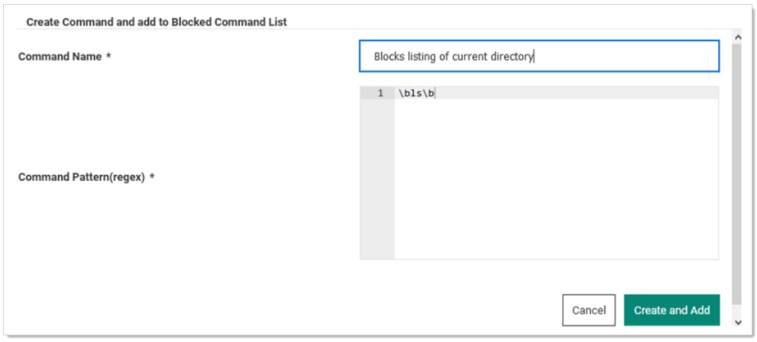


4. Click the **Create Blocked Command List** button. The New Blocked Command List page appears:



5. Type the list name in the **Name** text box.
6. Ensure the **Enabled** check box is selected.

7. **Either** select a predefined regex from the **Add Commands to Blocked Command List** dropdown list. **Or** add a custom regex of your own:
   a. Scroll to the bottom of the dropdown list.
   b. Click the **Create New Command** link. The Create Command and Add to Blocked Command List popup appears:

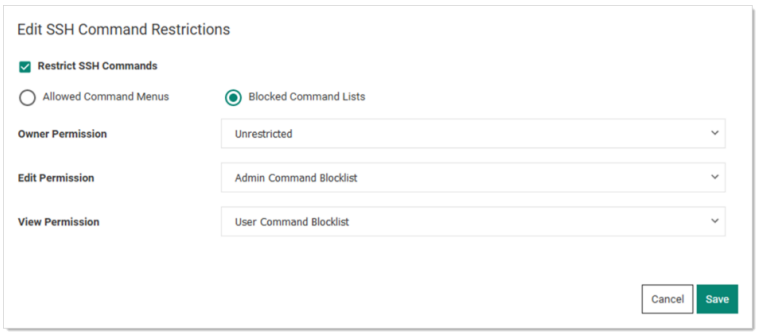


   c. Type the name for the command in the **Command Name** text box.
   d. Type or paste the regex in the **Command Pattern** text box.
   e. Click the **Create and Add** button. The command is added to the dropdown list.
8. Click the **Create Blocked Command** List button.

Applying SSH Command Blocked Lists in Secret Settings

To enable privilege management for an account:

1. Navigate to a PuTTY secret's **Settings** tab.
2. Go to the **SSH Launcher** section.
3. Click the **Edit** link.
4. Click the **Connect Using** dropdown list and select **Credentials on Another Secret**.
5. Click the **No Secret Selected** link to choose a secret containing your log on credentials, which the launcher uses when logging on the SSH service.
6. Enable command restrictions:
   a. Click the **Security** tab.
   b. Go to the **Other Security** section.
   c. Click the **Edit** link to set **Enable Proxy** to Yes.
   d. Click the **Edit** link for the **Enable SSH Command Restrictions**. The Edit SSH Command Restrictions popup appears:



e. Ensure the Restrict SSH Commands check box is selected.

f. Click to select the Blocked Command Lists selection button.

g. Click the **Owner**, **Edit**, and **View Permission** dropdown lists to map the blocked command lists to users via those permissions. You can also leave them as unrestricted.

h. Click the **Save** button.

SSH Command Restrictions via a Secret Policy

You can apply SSH command restrictions to a secret policy for ease of management. You can apply secret policies to secret folders or directly to a secret itself. To apply command restrictions, set a policy as follows:

Table: Secret Policy Security Settings for SSH Command Restrictions

Section	Item	Setting	Value
Security Settings	Enable Proxy	Enforced	Checked
Security Settings	Enable SSH Command Restrictions	Enforced	Checked
Security Settings	SSH Command Restriction Type	Enforced	Blocked List
Security Settings	SSH Command Blocklist for Secret Owners	Enforced or Not Set	Desired Block Command List or Not Set
Security Settings	SSH Command Blocklist for Secret Editors	Enforced or Not Set	Desired Block Command List or Not Set
Security Settings	SSH Command Blocklist for Secret Viewers	Enforced or Not Set	Desired Block Command List or Not Set

SSH Command Menus

Secret Server supports privilege management and command restrictions for UNIX and other platforms with SSH interfaces. Privilege management is an additional layer of access control that can be applied to secrets with SSH

Launchers over SSH Proxy. Privilege management gives the ability to grant users access to a machine with specific command restrictions to define the available commands that a user may run as root or another privileged account.

To use command restrictions, Secret Server must have SSH Proxy enabled.

With command menus, you can configure predefined commands that users or groups will be able to access when connecting as a privileged account. A command menu is a list of command names mapped to system commands. The format for specifying a command is to separate a name and command with an equals symbol. For example:

```
restart_apache = /usr/sbin/service apache restart
```

You may also use parameters in commands so users can execute more complex commands. For example:

```
move_file = /bin/mv $src $dst
```

You can specify environmental variables by escaping dollar signs in commands. For example:

```
go_home = cd $$HOME
```

Command restrictions currently do not support complex commands, such as multiple commands on one line, piping, or output redirection. To support these functions, you may add a script to the system that has these capabilities and point map the command to that script. We highly recommend that generated scripts have proper user permissions and that the absolute path is used. The absolute path ensures that the correct script is being executed.

Commands may not be named as numbers or one of the following predefined commands:

```
..
up
-help
?
-more
logout
exit
```

To enable privilege management for an account, navigate to a PuTTY Secret's launcher tab and specify the "Connect As" secret that you wish to connect as. When launching this secret, the launcher uses it as credentials to log into the SSH service and then log into the credentials specified on the secret.

To enable command restrictions, navigate to a PuTTY secret's security tab and specify "Enable Proxy" and "Enable SSH Command Restrictions." This gives you the ability to map users and groups to command menus. When the unrestricted command menu is specified for a user, the user is launched into a normal shell environment without command restrictions. Likewise, if the "Allow Owners Unrestricted SSH Commands" option is enabled, the owners of the secret are also launched into a normal shell environment without command restrictions.

When specifying command menus on a secret, at least one command menu must be selected unless "Allow Owners Unrestricted SSH Commands" is enabled.

You can apply command restrictions to a secret policy for ease of management.

SSH IP Block Listing

Introduction

SSH IP block listing is a feature that reduces the attack surface of the Secret Server SSH proxy and SSH terminal services.

Enabled by default, the feature adds to a block list any single IP address that fails to authenticate a number of connections across a defined time period. Once on the block list, the client IP is rejected when initiating an SSH connection, reducing the chances of a successful brute-force credential or denial of service (DOS) attack.

SSH Block Listing Rules

Client Override Settings

Client override settings allow creating allow/block lists, and blocked IP administration is provided via IP restriction administration.

Examples:

- A blocked IP is allowed if in a bypass allow-list entry
- A normally unblocked IP is blocked if included in a block list entry

Default Rules

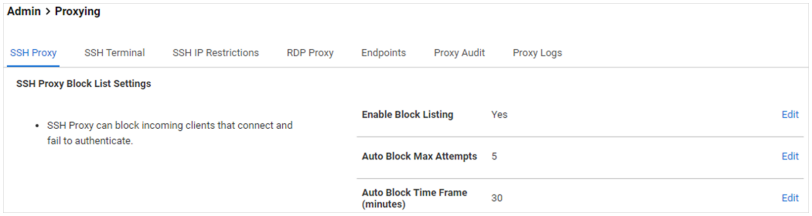
The defaults are:

- SSH block listing is enabled
- Five connection attempts are allowed in a 30-minute period
- The amount of history kept per client can be adjusted but must be greater than or equal to the Max Attempts

That is, any single IP that fails to authenticate five times in any one 30 minute period is added to a permanent block list.

SSH Proxy Block List Settings

The feature is administered via **Admin > Proxying > SSH Proxy**:



SSH proxy block-list settings:

- **Enable Block Listing:** Enable or disable the SSH block listing feature. When enabled, IPs reaching the defined blocking threshold are automatically added to the SSH IP restrictions block list. When disabled, all IP addresses

are able to connect and attempt to authenticate an unrestricted number of times.

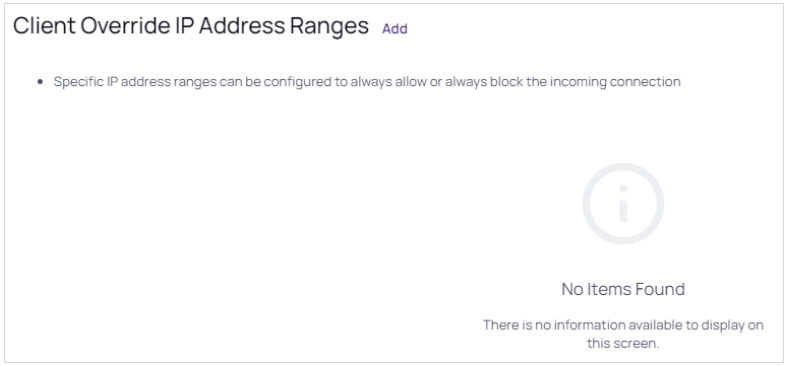
- **Auto Block Max Attempts:** The maximum failed or unauthenticated connections allowed.
- **Auto Block Max History:** The maximum number of attempts to keep in each clients history.
- **Auto Block Time Frame (minutes):** The period length during which the "Auto Block Max Attempts" must reach before a client IP is added to the block list.

Client Override IP Address Ranges

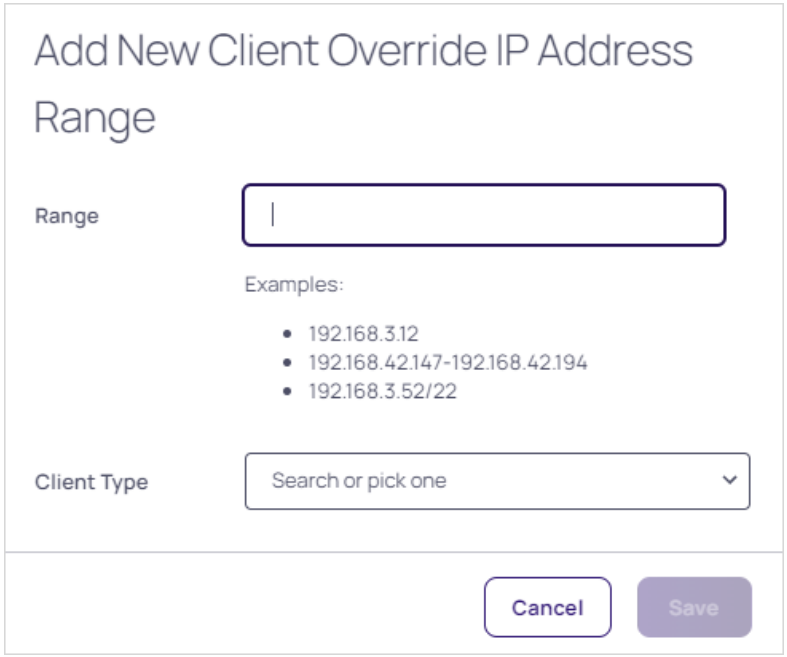
You can configure specific IP address ranges to always allow or always block an incoming connection. If allowed, authentication is still required to access the SSH proxy or SSH terminal services.

To add a range:

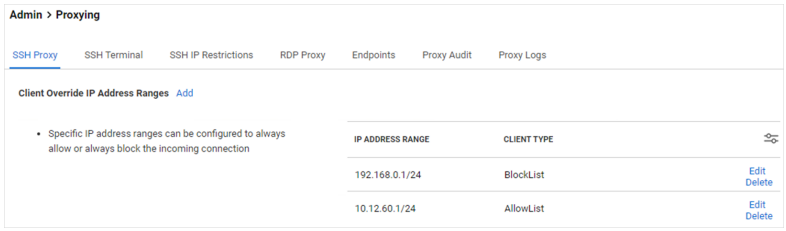
1. Scroll down on the same tab:



2. Click the **Add** link. A popup appears:



3. Type an IP address or range in the **Range** text box. You can use CIDR notation. Examples:
   - 192.168.3.12
   - 192.168.42.147-192.168.42.194
   - 192.168.3.52/22
4. Click the **Client Type** dropdown list to select **Allow List** or **Block List**.
5. Click the **Save** button. Your choices appear:

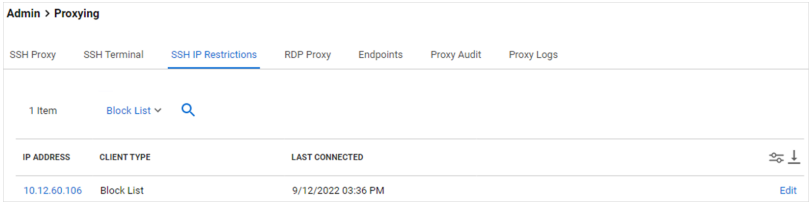


These rules have priority over individual IP client settings in the SSH IP restrictions list.

IP Address Management

SSH IP Restrictions

Client IP address management is accessed via **Admin > Proxying > SSH IP Restrictions**:



This page lists client IP addresses that were added by a connection exceeding the limits allowed for failed or unauthenticated attempts to one of the proxy endpoints, including any that have been previously reclassified.

The list has a number of built in display filters:

- All
- Allow List
- Block List
- Unknown

Use the search feature to locate specific IP addresses.

Managing IP Addresses

If an IP address is located on the SSH IP Restrictions page, you can view or edit its current SSH proxy block status. For example, a user may have inadvertently blocked her client IP for a number of reasons, such as testing the

connection too many times in a short time period or performing a vulnerability or port scan against the proxy endpoint IP addresses.

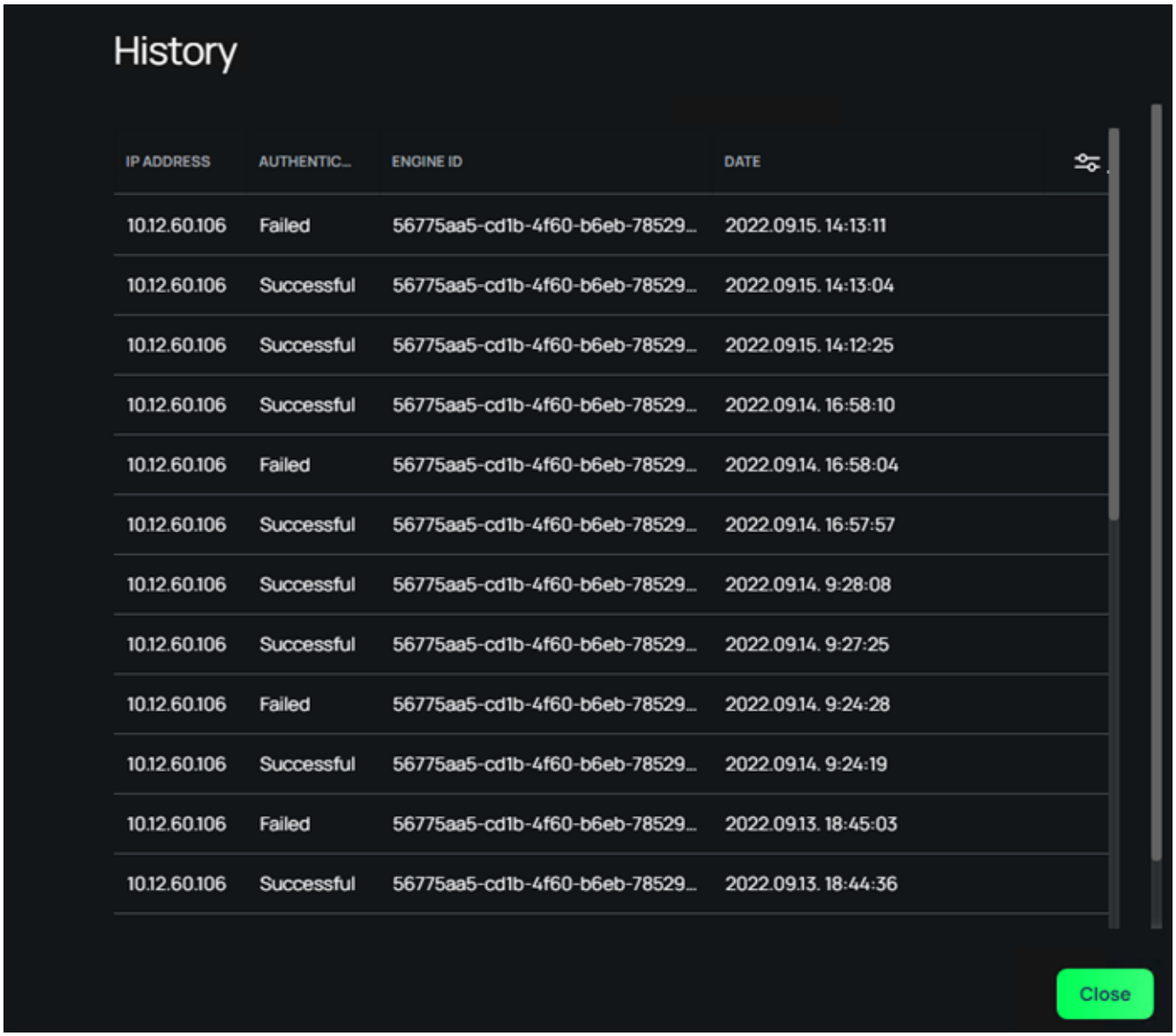
Client Types

The SSH IP Restrictions page categorizes clients as follows:

Client Type	Access	Rule Type	Description	Notes
Allow List	Yes	Static	These IPs are allowed. Meeting the automatic blocking thresholds will not result in block listing.	Could be useful to allow list a vulnerability scanner. Is overridden by a matching override block list.
Block List	No	Static	These IPs are always blocked.	Is overridden by a matching override allow list.
Unknown	Yes	Dynamic	These are IPs that have previously accessed the SSH proxy and have not met the automatic block thresholds. They are in neither static allow nor block states	This is a client default, even if covered by a override allow or block list.

IP Address Activity

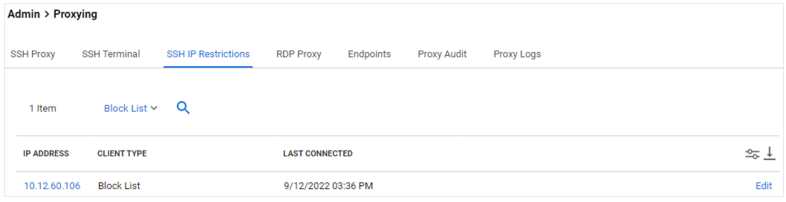
The SSH IP Restrictions page allows you to view the connection history for an IP address:



This shows the authentication status of each connection attempt, which is especially helpful when troubleshooting SSH terminal connectivity.

Blocking or Unblocking Client IPs

1. Go to the **SSH IP Restrictions** tab:



2. Locate the desired IP address in the list using the filter and search features.
3. Click the **Edit** link to the right of that IP address in the list. A popup appears.
4. Click the **Client Type** dropdown list to select either **Allow List** or **Block List** for the client type.
5. Click the **Save** button.

Troubleshooting

If a user reports he or she is unable to connect an SSH proxied or tunneled secret launcher for IPs included in the block list, you will see the following logging:

Secret Server On-Premises

For Secret Server On-Premises where the SSH proxy is on a web-node (IIS):

Path: `c:\inetpub\wwwroot\SecretServer\log\SS.log`

Message:

```
INFO Thycotic.SSHProxy.SSHServer - SSHProxy_Server_ConnectionReceivedINFO
Thycotic.SSHProxy.Logic.ConnectionBridge - SSHProxy_Client_Stopped_NoAuthWARN
Thycotic.SSHProxy.Logic.ConnectionBridge - SSH Proxy host 10.12.60.148:22 refused a
connection from client 10.12.60.106:52572 for too many failed authentication attempts. If
this was in error, please go to Admin > Proxying, click the SSH IP Restrictions tab and
update the Client Type to Allow List for this client IP.
```

Distributed Engines

For Secret Server Cloud or On-Premises with a distributed engine:

Path: `C:\ Program Files\Thycotic Software Ltd\Distributed Engine\log\SSDE.log`

Message:

```
INFO Thycotic.SSHProxy.SSHServer - SSHProxy_Server_ConnectionReceived - String[]
{10.12.60.148:22, 10.12.60.106:50182}INFO Thycotic.SSHProxy.Logic.ConnectionBridge -
SSHProxy_Client_Stopped_NoAuth - String[] {}~WARN Thycotic.SSHProxy.Logic.ConnectionBridge -
SSHProxy_Server_ConnectionRefused - (null)INFO
Thycotic.SSHProxy.Logic.ClientToServerConnection - ClientToServerConnection, buffer, and
tunnel disposed - (null)
```

SSH Jumpbox Routes

An *SSH jumpbox route*, is a series of regular Linux servers, accessible from the Internet, that is a gateway to other Linux machines on a private network using the SSH protocol. This topic and its subtopics address discuss using jumpbox routes.

> SSH jumpboxes are also called *bastion hosts*, *jump hosts*, or *jump box servers*. *Bastion* is a military term meaning *a projecting part of a fortification*. Bastion hosts are hardened and monitored servers that reside outside of an organization's security zone, usually exposed to the internet. All jumpboxes are bastion hosts, but all bastion hosts are not necessarily jumpboxes.

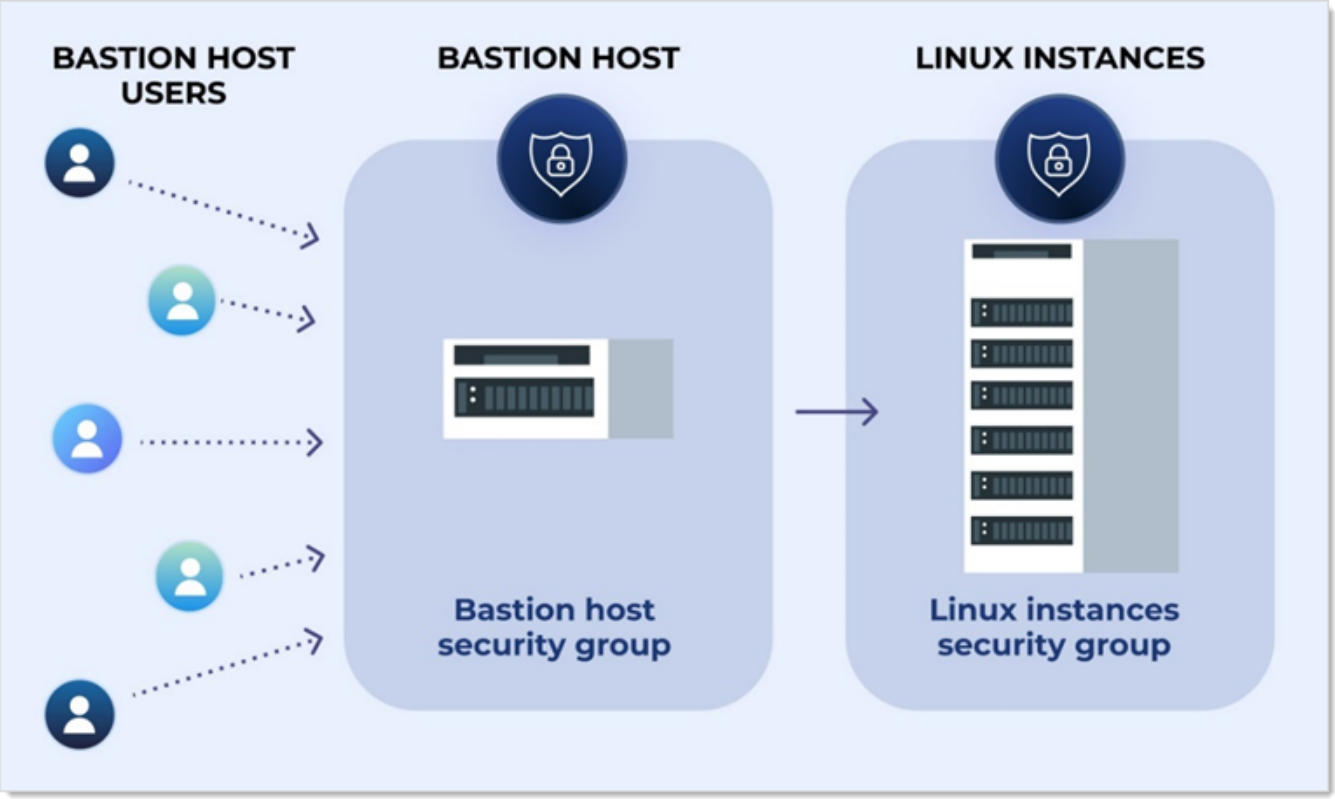
Because SSH jumpboxes usually reside on the Internet, they run a minimum of services to reduce their attack vulnerability. Similarly, limiting the Internet access to your infrastructure to one hardened gateway server also

reduces risk. In addition, a dedicated SSH access point makes it easier to have an aggregated audit log of all SSH connections.

With early SSH, users had to SSH into a jump host and then type `ssh` again to manually jump to a destination host. Today, this is done automatically using the built-in SSH `-J` ProxyJump option.

Secret Server can now create a chain of jumpbox secret connections to reach an otherwise inaccessible Linux instance. This sequence is called a *jumpbox route* and can contain up to 20 jumpbox levels (hops between instances).

Figure: SSH Jumpbox Route Setup



Best Practices for Jumpbox Routes

See "SSH Jumpbox Routes" on the previous page for an introduction.

Overview

This topic discusses configuring an SSH jumpbox route using OpenSSH.

We strongly recommend a dedicated SSH jumpbox without any other publicly accessible software on the machine. Occasionally, multiple jumpboxes (a jumpbox route) may be required to access a target machine that is behind several layers of network security.

Assumptions

We want:

- The jumpbox to forward SSH connections to our internal hosts. Users must authenticate to the SSH jumpbox and the internal target host. Users need valid credentials for both hosts.
- A single shared user for everyone with no interactive terminal sessions allowed.
- Users to connect to internal target hosts using Secret Server's jumpbox route feature, which uses the jumpboxes as jump route "levels" to ultimately launch a target secret that has this jumpbox route selected.

Best Practices

We recommend not allowing users to log into a jumpbox directly because they might introduce security issues by:

- Inadvertently updating the jumpbox configuration
- Using the jumpbox machine for unrelated tasks.
- Making copies of cryptographic keys used to access destination servers.

We recommend changing the default TCP port on the SSH jumpbox from 22 to something else.

Example Setup

The following example uses these conventions:

- The organization domain is example.com.
- The DNS name of the jumpbox is proxy.example.com, which is the only machine accessible from the Internet.

Task 1: Launching a New Linux Instance

- Stand up a Linux instance on your cloud provider. We use Ubuntu 20.04 LTS because it is simple, well-supported, and includes the recently released OpenSSH 8.2.
- Set up a firewall or security group policy to restrict connections to the jumpbox to port 22 (SSH).
- Allow connections only from IPs you trust.

Task 2: Configuring OpenSSH

- Read and apply Mozilla's OpenSSH Security Guidelines.

  
  The guidelines only cover up to OpenSSH 6.7. Most are still relevant to OpenSSH 8.2.

- OpenSSH is bundled by default with most Linux distributions. It is almost certain your Linux machine already has it installed. If the server is accessible via proxy.example.com, clients can access other servers behind the same NAT boundary via the `-J` command line flag. For instance: `$ ssh -J proxy.example.com 10.2.2.1`

  In the example above, the jumpbox is used to access another host on an AWS VPC with an address of 10.2.2.1.

Task 3: Configuring Public Key Authentication and Disabling Root Logins

Configuring the sshd_config File

In your new jumpbox's `/etc/ssh/sshd_config` file, consider adding the following SSHD config parameters:

```
# Supported HostKey algorithms by order of preference.
HostKey /etc/ssh/ssh_host_ed25519_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_rsa_key
# Password based logins are disabled - only public key based logins are allowed.
AuthenticationMethods publickey
# LogLevel VERBOSE logs user's key fingerprint on login. Needed to have a clear audit
track of which key was using to log in.
LogLevel VERBOSE
PermitRootLogin no
# Log sftp level file access (read/write/etc.) that would not be easily logged otherwise.
Subsystem sftp  /usr/lib/ssh/sftp-server -f AUTHPRIV -l INFO
```

The setup above works only when the public SSH keys are properly distributed, not only between clients and the jumpbox but also between the clients and the destination servers. With these settings, password authentication is disabled.

Setting Key Exchange Algorithms, Ciphers, and MACs

Consider which algorithms and key types you want to support. Mozilla recommends the following key types (more restrictive than the OpenSSH defaults):

In `/etc/ssh/sshd_config`, add:

```
KexAlgorithms curve25519-sha256@libssh.org,ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-
sha2-nistp256,diffie-hellman-group-exchange-sha256
Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-
gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,umac-128-
etm@openssh.com,hmac-sha2-512,hmac-sha2-256,umac-128@openssh.com
```

Task 4: Disabling Forwarding

Next, harden the server configuration by disabling interactive SSH sessions on the jumpbox for regular users, leaving it turned on for the administrators. To do this, update the sshd configuration `/etc/ssh/sshd_config` with the following:

```
# Do not let SSH clients do anything except be forwarded to the destination:
AllowAgentForwarding no
AllowStreamLocalForwarding no
X11Forwarding no
```

If you only have a few users, consider creating a separate user account on the jumpbox dedicated to “jumping users.” For example, call it "jumpuser" and update the configuration `/etc/ssh/sshd_config`:

```
Match User jumpuser
```

```
PermitTTY no
X11Forwarding no
PermitTunnel no
GatewayPorts no
```

Task 5: Changing the Default SSH Port

If you do not want to restrict access by IP address in your security group rules, consider changing your default SSH port. This will deter many basic bots.

In `/etc/ssh/sshd_config`, add `Port 37271`.

Task 6: Testing Configuration and Restarting SSHD Service

You can test your configuration with `sshd -t`, then restart the SSHD server. Make sure you can still ssh into the machine before you continue!

Task 7: Send Your Users' SSH Keys to the Jumpbox

Add the jump user's public keys to the `/home/jumpuser/.ssh/authorized_keys` file. These keys can be stored in Secret Server and used during the Jumpbox Route launch session.

Using Bastion Hosts with Secret Server and Jumpbox Routes

Please see "Creating and Testing Secrets for Jumpbox Routes" on page 871.

References

Johnson, Brian. "How to Create a Bastion Host | Part 1 of a Step-by-step Tutorial." *Strongdm*, 25 March, 2021, https://www.strongdm.com/blog/bastion-hosts-with-audit-logging-part-one, Accessed 12 October, 2021

Kontsevoy, Ev. "How to Set Up an SSH Jump Server." *Teleport*, 30 March, 2021, https://goteleport.com/blog/ssh-jump-server/, Accessed 12 October, 2021

Tashian, Carl. "DIY SSH Bastion Host." *Small step*, 08 July 2020, https://smallstep.com/blog/diy-ssh-bastion-host/, Accessed 12 October, 2021

Creating and Editing SSH Jumpbox Routes

See "SSH Jumpbox Routes" on page 863 for an introduction.

Introduction

You can add jumpbox routes to Unix secrets when SSH proxying is enabled. A jumpbox route assigned to a target secret specifies a series of Unix servers called *jumpboxes*, *jumpbox servers*, or *bastion hosts*. The proxied connection is forwarded through the jumpbox route before reaching the server specified in the target secret.

Prerequisites

Using jumpbox routes requires these prerequisites:

- Only use Unix servers and secrets: At this time, jumpbox routes are only supported by Unix secrets. The option to add a jumpbox route does not appear on any target secret that is not base on the Unix secret template. Similarly, a jumpbox route that attempts to connect to jumpbox servers that do not use a Unix secret template will also fail.
- Enable SSH proxy on Secret Server: For more information on how to enable proxying on Secret Server, see "SSH Proxy Configuration" on page 831.
- Enable the SSH Proxy setting for the target secret using a jumpbox route: To assign a jumpbox route to a target secret, the target secret itself must have SSH proxying enabled. The enable SSH Proxy check box appears in the secret's Security tab if proxying is enabled on Secret Server itself.

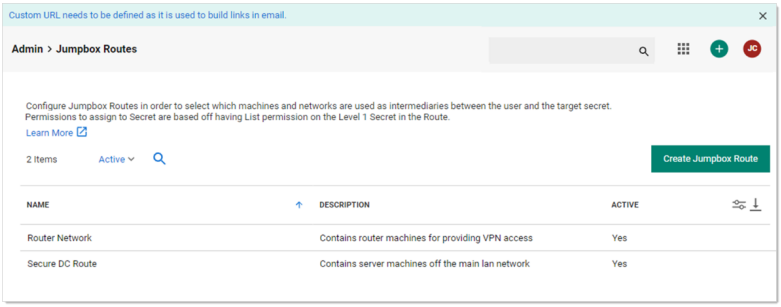
Permissions

Using jumpbox routes requires these permissions:

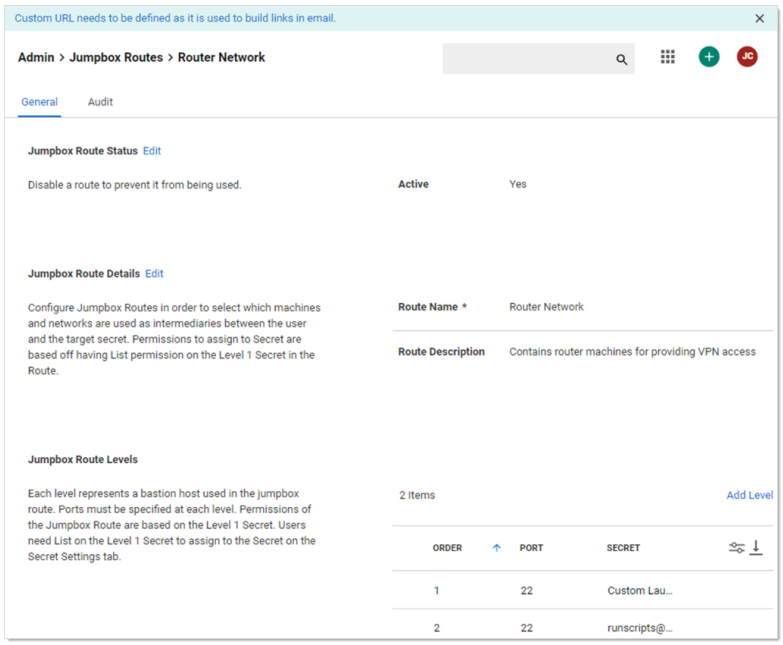
- Adding a jumpbox route to a target secret: A user must have owner permissions on a secret to assign, change, or remove that secret's jump server route. Additionally, users are only able to pick from a list of routes where they have at least list permission on the first jump route server.
- Editing Jumpbox Routes: Users must have the "Administer Jumpbox Route" permission to create, edit, or deactivate jump server routes. Users with the "View Jumpbox Route" permission can view the details of all jump server routes in the Admin Jumpbox Route page, but they cannot make any changes.

Creating, Editing, and Deactivating Jumpbox Routes

1. Go to **Admin > See All**.
2. Search for and select **Jumpbox Routes**. The Jumpbox Routes page appears:



   This page lists all existing jumpbox routes.
3. Select an existing route to edit or create a new route by clicking the **Create Jumpbox Route** button.
4. If you chose to create a new route, a popup appears. Type the name and description and click the **Create Jumpbox Route** button.
5. In either case, the Jumpbox Routes page appears for the new or edited route:



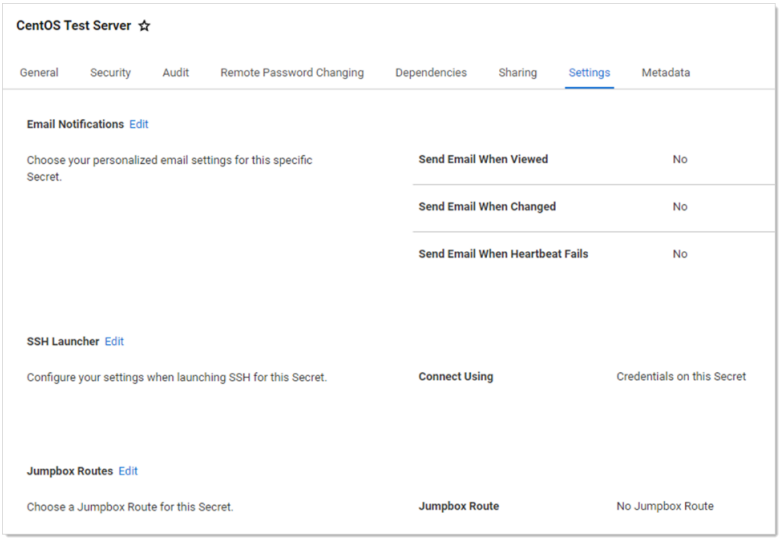
6. Note that routes can only be deactivated, not deleted, in the **Jumpbox Route Status** section.
7. You can view or edit the route's jump servers in the **Jumpbox Route** Levels section. Each row in the grid represents a *level*, each of which has these attributes:
   - Secret: Each level contains one secret that represents a jump server.
   - Port: The port used on the jump server for the route. Choose an available port on which the server listening for SSH connections. Typically, this is port 22, but we recommend changing it to deter basic script bots, see "Best Practices for Jumpbox Routes" on page 864 for more information.

     This value overrides any port specified on the secret.

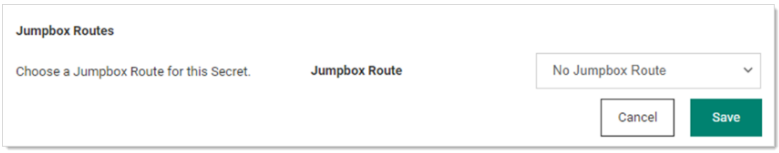
   - Order: The sequence of jumpboxes forwarding the proxied connection, starting with 1. You can hover over any level to present a handle for dragging it to reorder the list.
8. Click the **Add Level** hyperlink to add a jumpbox to the route. An Add Level popup appears.
9. Select or type your desired port in the **Port** list box.
10. Click the **No Secret Selected** hyperlink to choose the secret for the new level (server).
11. Click the **Save** button.
12. Hover over any level to present a handle (on the left) for dragging the level to reorder the list.
13. Hover over any level to present an ellipsis icon (on the right).
14. Click the icon and choose **Edit** or **Delete** to change or remove the level. You can change the port or related secret. Deletion permanently removes the level from the jumpbox route. It is not deactivated as with the route itself.

Assigning a Jumpbox Route to a Secret

1. Select or create a Unix secret for the route. Ensure that proxying and all other prerequisites described above are met.
2. Navigate to the secret's **Settings** tab and go to the **Jumpbox Route** section.



3. Click the **Edit** hyperlink. The section becomes editable:



4. Click to select the desired jumpbox route from the drop-down. Only active jump server routes to which you have access are visible. You can unassign the jumpbox route by selecting **No Jumpbox Route** in the dropdown list.

> If the jumpbox route becomes deactivated, "Router Network (inactive)" appears in the Jumpbox Route dropdown list. You should select a new one.

> You can also assign jumpbox routes though a secret policy. See "Creating Secret Policies" on page 1127.

Global Jumpbox Settings

The **Admin > Proxying** page has these jumpbox settings that impact your jumpbox routes:

- **Tunnel Keep Alive:** This setting is to stop intermediaries (such as firewalls or proxies) from deeming your connections inactive and timing them out and closing them. The default is 50 seconds (just shy of a common timeout setting). Adjust this to be just short of your network's timeout setting. Type 0 to disable the setting.

- **Available Port Range:** This sets the port range for the SSH proxy endpoint for jumpbox forwarding. Type a range separated by a hyphen, such as 10000-15000. The ports are used for local port forwarding for jumps. You cannot use the ports for other processes, so we recommend using a high range (ports above 10000).

Creating and Testing Secrets for Jumpbox Routes

Overview

This topic shows how to create secrets used for jump server routes by way of an extended example.

> Please see "Best Practices for Jumpbox Routes" on page 864 for creating hardened jump servers for use in jump server routes.

We use following scenario:

A target Ubuntu host is only accessible via an SSH `proxyjump` directive. The sequence is Secret Server to `bastion01.thycotic.com` to `bastion02.thycotic.com` to `remote-hostname.thycotic.com`, the target Ubuntu host.

As suggested in the **Bastion Design Best Practices** guide:

- Both bastion hosts 01 and 02 have changed their default SSH ports for all incoming connections to 2222 and 3333, respectively.
- Both use a single jump user, betty on bastion01 and wilma on bastion02.
- Both users have SSH public key authentication enabled and have an `/home/<username>/.ssh/authorized_keys` file storing all the public keys.

You can connect to the target server without Secret Server from any SSH client using the following SSH command, with the built-in SSH ProxyJump directive -J and defined aliases:

```
$ ssh -J <jumpbox1>, <jumpbox2> <remote-host-target>
```

...with local `~/.ssh/config` aliases configured this way:

```
### First Jumpbox
Host jumpbox1
  HostName bastion01.thycotic.com
  Port 2222
  User betty
  IdentityFile /home/betty/.ssh/jumpbox1_ssh_rsa
### Second Jumpbox
Host jumpbox2
  HostName bastion02.thycotic.com
  Port 3333
```

To replicate this jump scenario using Secret Server, we need to create three new secrets and a new jumpbox route, and then assign that route to the target secret. Once that is done, launching the remote-host-target secret is as simple as clicking the PuTTY launcher icon–the jumpbox route is automatically set up for you.

Task 1: Creating Jumpbox and Target Host Secrets

All three connections require SSH public key authentication.

1. Create a new secret based on the Unix Account (SSH Key Rotation) template with these particulars:
   - Secret name: bastion01
   - Machine: bastion01.thycotic.com
   - Username: betty
2. Upload or generate the private/public key pair. Set a passphrase if necessary.
3. If the key pair is generated, save the public key on the secret to the user's `/home/betty/.ssh/authorized_keys` file on the jumpbox.
4. Create a second new secret also based on the Unix Account (SSH Key Rotation) template with these particulars:
   - Secret name: bastion02
   - Machine: bastion02.thycotic.com
   - Username: wilma
5. Upload or generate the private/public key pair. Set a passphrase if necessary.
6. If the key pair is generated, save the public key on the secret to the user's `/home/wilma/.ssh/authorized_keys` file on the jumpbox.
7. Create a third new secret also based on the Unix Account (SSH Key Rotation) template with these particulars:
   - Secret name: remote-host-target
   - Machine: remote-hostname.thycotic.com
   - Username: remote_user
8. Upload or generate the private/public key pair. Set a passphrase if necessary.
9. If the key pair is generated, save the public key on the secret to the user's `/home/remote_user/.ssh/authorized_keys` file on the target host.

Task 2: Creating a Jumpbox Route with Secret Levels

See "Creating and Editing SSH Jumpbox Routes" on page 867 for more on creating jumpbox routes.

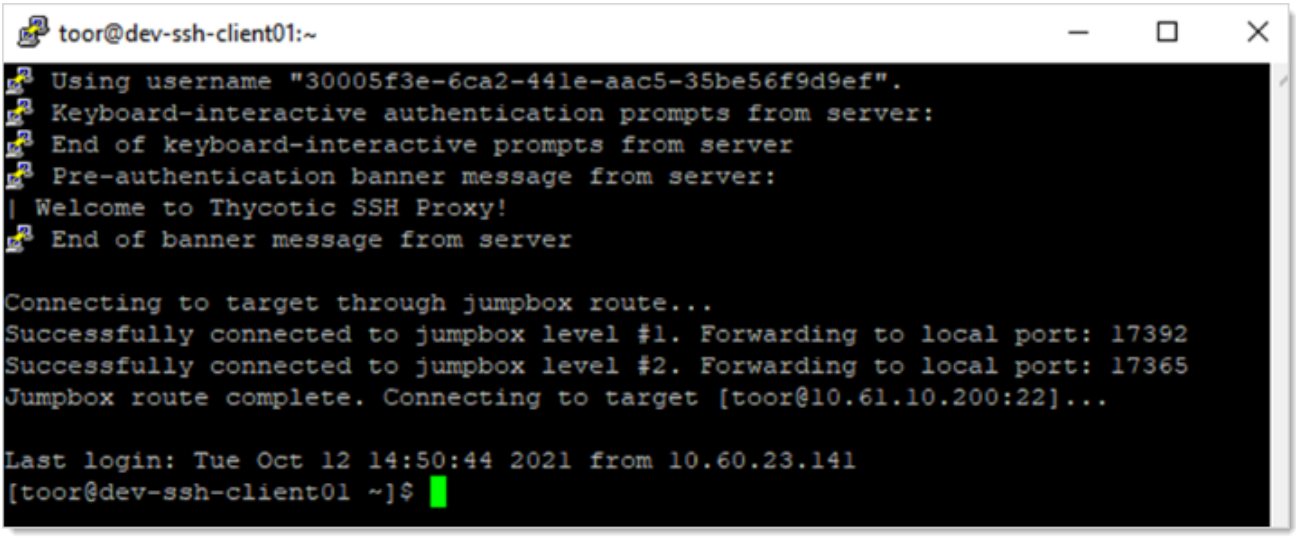
1. Create a new jumpbox route named flintstone.
2. For level 1, select the bastion01 secret and choose 2222 as the port.
3. For level 2, Select the bastion02 secret and choose 3333 as the port.

Task 3: Assigning a Jumpbox Route on the Target Secret

1. Open the remote-host-target secret.
2. Ensure SSH proxying is enabled (Security > Enable SSH Proxy) on the secret.
3. Navigate to the secret's **Settings** tab.
4. Scroll down to edit the **Jumpbox Route** section.
5. Select the **Flintstone Route** in the drop-down.
6. Save your changes.

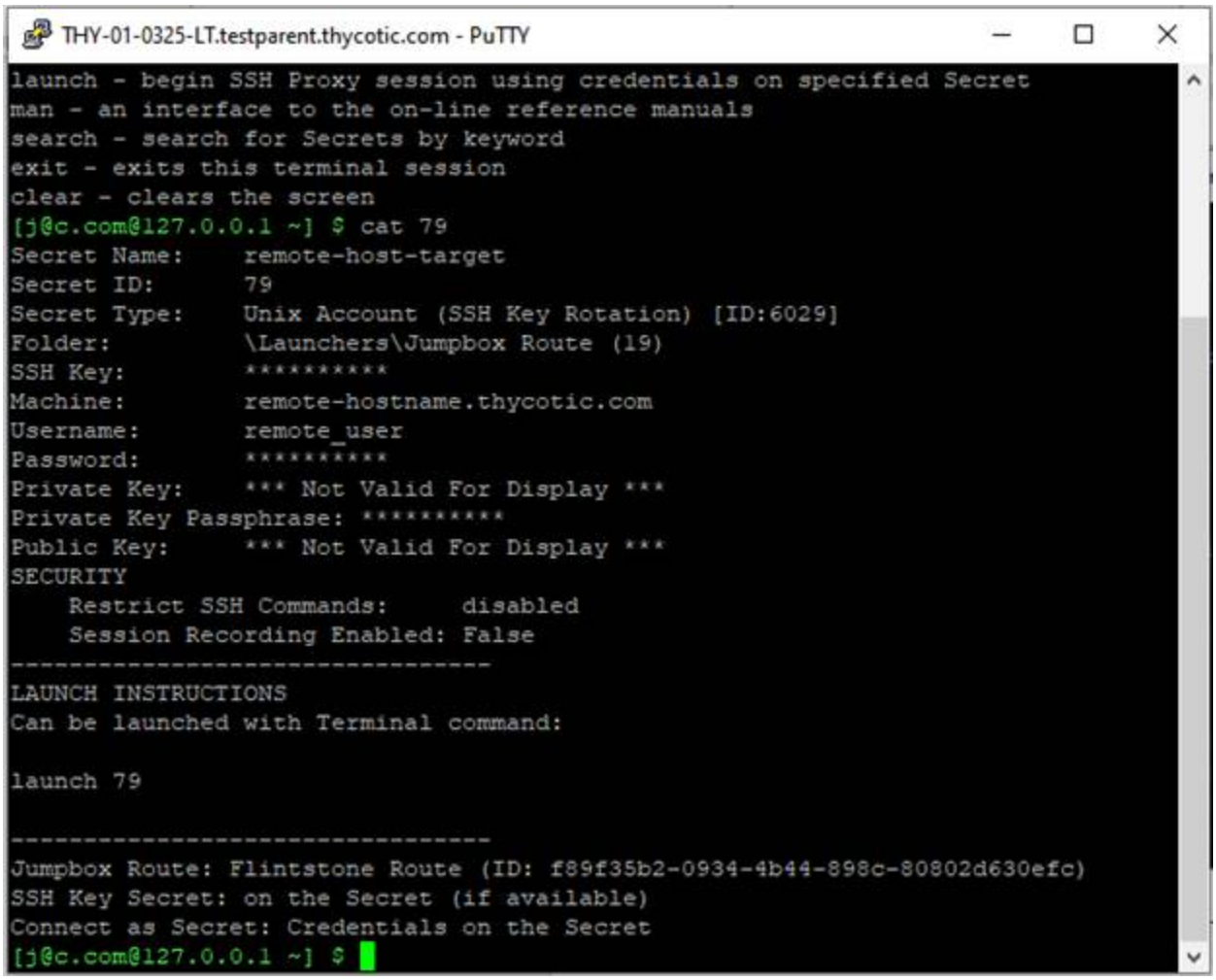
Task 4: Test Launching the Target Secret with PuTTY Launcher over SSH Proxy

1. Open the remote target secret.
2. Click the **General** tab.
3. Click the **PuTTY Launcher** link to begin a new SSH proxy session with a jump server route. The launcher connects to the target via the jump server route:

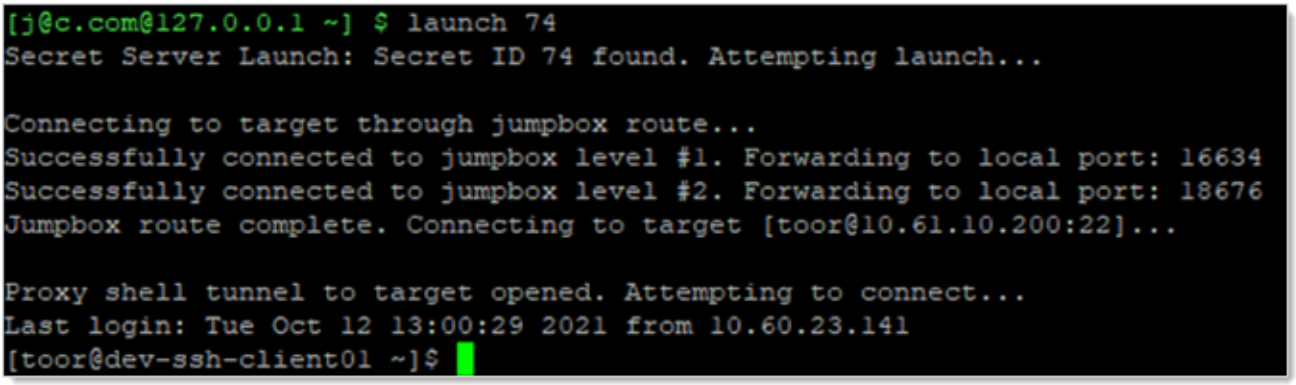


Task 5: Test Launching the Target Secret with SSH Terminal

1. Note the secret ID of your remote target secret.
2. Using an SSH client, log in to SSH Terminal. See "SSH Terminal Administration" on page 834 for details.
3. Enter `cat <secret id>` to view the remote target secret's details:



4. Enter `launch <secret id>` to launch the remote target secret with a jumpbox route:



5. You should now be connected to remote-hostname.thycotic.com as remote_user.
6. Enter `exit` to return to the terminal.

SSH Cipher Support

This topic details SSH cipher suite encryption, key exchange, and MAC algorithms.

Enable FIPS in Secret Server On-Premises to ensure all algorithms are FIPS-certified. FIPS 140-2 compliance is built-in to Secret Server Cloud and is always on.

SecureBlackbox enables all available SSH encryption, key exchange, and MAC algorithms by default.

This information applies to the following as of Secret Server On-Premises 11.2.X (June 2022).

- SSH Server: Used by SSH proxy
- SSH Client: Used by SSH proxy, RPC, heartbeat, discovery, and script runners.
- Local port forwarding: Used by SSH proxy Jumpbox routes)

Secret Server On-Premises with FIPS Enabled

Default Encryption Algorithms, FIPS

The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
aes256-cbc	SSH_EA_AES256	2147483646
aes192-cbc	SSH_EA_AES192	2147483645
aes128-cbc	SSH_EA_AES128	2147483644
3des-cbc	SSH_EA_3DES	2147483643

Default Key Exchange Algorithms, FIPS

The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
rsa1024-sha1	SSH_KEX_RSA1024_SHA1	2147483646
rsa2048-sha256	SSH_KEX_RSA2048_SHA256	2147483645

Default MAC Algorithms, FIPS

The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
hmac-sha2-512	SSH_MA_HMAC_SHA2_512	2147483646
hmac-sha2-256	SSH_MA_HMAC_SHA2_256	2147483645
hmac-sha256@ssh.com	SSH_MA_HMAC_SHA256	2147483644
hmac-sha256-96@ssh.com	SSH_MA_HMAC_SHA256_96	2147483643
hmac-sha1	SSH_MA_HMAC_SHA1	2147483642

Default Public-Key Algorithms, FIPS

The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
x509v3-sign-rsa	SSH_PK_X509_SIGN_RSA	2147483646
x509v3-sign-dss	SSH_PK_X509_SIGN_DSS	2147483645
spki-sign-rsa	SSH_PK_SPKI_SIGN_RSA	2147483644
spki-sign-dss	SSH_PK_SPKI_SIGN_DSS	2147483643
pgp-sign-rsa	SSH_PK_PGP_SIGN_RSA	2147483642

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
pgp-sign-dss	SSH_PK_PGP_ SIGN_DSS	2147483641
x509v3-ssh-rsa	SSH_PK_X509_ SSH_RSA	2147483640
x509v3-ssh-dss	SSH_PK_X509_ SSH_DSS	2147483639
x509v3-rsa2048-sha256	SH_PK_X509_ RSA2048_SHA256	2147483638
rsa-sha2-256	SSH_PK_RSA_ SHA256	2147483637
rsa-sha2-512	SSH_PK_RSA_ SHA512	2147483636
ssh-dss	SSH_PK_DSS	2147483635
ssh-rsa	SSH_PK_RSA	2147483634

Secret Server with FIPS Disabled

Default Encryption Algorithms, Non-FIPS

The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
aes256-gcm@openssh.com	SSH_EA_AES256_GCM_OPENSSH	2147483646
aes128-gcm@openssh.com	SSH_EA_AES128_GCM_OPENSSH	2147483645
aes256-gcm	SSH_EA_AES256_GCM	2147483644
aes128-gcm	SSH_EA_AES128_GCM	2147483643
aes256-ctr	SSH_EA_AES256_CTR	2147483642
aes192-ctr	SSH_EA_AES192_CTR	2147483641

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
aes128-ctr	SSH_EA_AES128_CTR	2147483640
aes256-cbc	SSH_EA_AES256	2147483639
aes192-cbc	SSH_EA_AES192	2147483638
aes128-cbc	SSH_EA_AES128	2147483637
3des-cbc	SSH_EA_3DES	2147483636
twofish256-cbc	SSH_EA_TWOFISH256	36
twofish192-cbc	SSH_EA_TWOFISH192	35
twofish128-cbc	SSH_EA_TWOFISH128	34
serpent256-cbc	SSH_EA_SERPENT256	33
serpent192-cbc	SSH_EA_SERPENT192	32
serpent128-cbc	SSH_EA_SERPENT128	31
blowfish-cbc	SSH_EA_BLOWFISH	30
twofish128-ctr	SSH_EA_TWOFISH128_CTR	29
twofish192-ctr	SSH_EA_TWOFISH192_CTR	28
twofish256-ctr	SSH_EA_TWOFISH256_CTR	27
serpent128-ctr	SSH_EA_SERPENT128_CTR	26
serpent192-ctr	SSH_EA_SERPENT192_CTR	25
serpent256-ctr	SSH_EA_SERPENT256_CTR	24
blowfish-ctr	SSH_EA_BLOWFISH_CTR	23
idea-ctr	SSH_EA_IDEA_CTR	22
cast128-ctr	SSH_EA_CAST128_CTR	21
arcfour128	SSH_EA_ARCFOUR128	20
arcfour256	SSH_EA_ARCFOUR256	19

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
cast128-cbc	SSH_EA_CAST128	18
3des-ctr	SSH_EA_3DES_CTR	16
chacha20-poly1305	SSH_EA_CHACHA20	15
arcfour	SSH_EA_ARCFOUR	14
idea-cbc	SSH_EA_IDEA	13
chacha20-poly1305@openssh.com	SSH_EA_CHACHA20_OPENSSH	12
des-cbc	SSH_EA_DES	11
none	SSH_EA_NONE	10

Default Key Exchange Algorithms, Non-FIPS

The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
curve25519-sha256@libssh.org	SSH_KEX_CURVE25519	2147483646
diffie-hellman-group-exchange-sha256	SSH_KEX_DH_GROUP_EXCHANGE256	2147483645
diffie-hellman-group14-sha1	SSH_KEX_DH_GROUP_14	2147483644
diffie-hellman-group1-sha1	SSH_KEX_DH_GROUP	2147483643
diffie-hellman-group-exchange-sha1	SSH_KEX_DH_GROUP_EXCHANGE	2147483642
diffie-hellman-group14-sha256	SSH_KEX_DH_GROUP_14_SHA256	2147483641
ecdh-sha2-nistp521	SSH_KEX_ECDH_NIST_P521	2147483640
ecdh-sha2-nistp384	SSH_KEX_ECDH_NIST_P384	2147483639
ecdh-sha2-nistp256	SSH_KEX_ECDH_NIST_P256	2147483638
rsa1024-sha1	SSH_KEX_RSA1024_SHA1	2147483637
rsa2048-sha256	SSH_KEX_RSA2048_SHA256	2147483636

Default MAC Algorithms, Non-FIPS

The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
chacha20-poly1305@openssh.com	SSH_MA_POLY1305	2147483646
aes256-gcm	SSH_MA_AES256_GCM	2147483645
aes128-gcm	SSH_MA_AES128_GCM	2147483644
hmac-sha2-512	SSH_MA_HMAC_SHA2_512	2147483643
hmac-sha2-256	SSH_MA_HMAC_SHA2_256	2147483642
hmac-sha256@ssh.com	SSH_MA_HMAC_SHA256	2147483641
hmac-sha256-96@ssh.com	SSH_MA_HMAC_SHA256_96	2147483640
hmac-sha1	SSH_MA_HMAC_SHA1	2147483639
umac-128@openssh.com	SSH_MA_UMAC128	2147483638
umac-96@openssh.com	SSH_MA_UMAC96	2147483637
umac-64@openssh.com	SSH_MA_UMAC64	2147483636
umac-32@openssh.com	SSH_MA_UMAC32	2147483635
hmac-sha2-512-etm@openssh.com	SSH_MA_HMAC_SHA2_512_ETM	28
hmac-sha2-256-etm@openssh.com	SSH_MA_HMAC_SHA2_256_ETM	27
hmac-sha256-96@ssh.com	SSH_MA_HMAC_SHA256_96	24
hmac-ripemd160	SSH_MA_HMAC_RIPEMD160	23
hmac-ripemd	SSH_MA_HMAC_RIPEMD	22
hmac-ripemd160@openssh.com	SSH_MA_HMAC_RIPEMD_OPENSSH	21
hmac-sha1-96	SSH_MA_HMAC_SHA1_96	15
hmac-md5	SSH_MA_HMAC_MD5	13
hmac-md5-96	SSH_MA_HMAC_MD5_96	12

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
none	SSH_MA_NONE	10

Default Public-Key Algorithms, Non-FIPS

The algorithm with highest priority is chosen first, if unsuccessful, the next highest is attempted. The table is ordered by priority.

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
x509v3-sign-rsa	SSH_PK_X509_SIGN_RSA	2147483646
x509v3-sign-dss	SSH_PK_X509_SIGN_DSS	2147483645
spki-sign-rsa	SSH_PK_SPKI_SIGN_RSA	2147483644
spki-sign-dss	SSH_PK_SPKI_SIGN_DSS	2147483643
pgp-sign-rsa	SSH_PK_PGP_SIGN_RSA	2147483642
pgp-sign-dss	SSH_PK_PGP_SIGN_DSS	2147483641
ecdsa-sha2-nistp256	SSH_PK_ECDSA_NIST_P256	2147483640
ecdsa-sha2-nistp384	SSH_PK_ECDSA_NIST_P384	2147483639
ecdsa-sha2-nistp521	SSH_PK_ECDSA_NIST_P521	2147483638
ecdsa-sha2-nistk163	SSH_PK_ECDSA_NIST_K163	2147483637
ecdsa-sha2-nistp192	SSH_PK_ECDSA_NIST_P192	2147483636
ecdsa-sha2-nistp224	SSH_PK_ECDSA_NIST_P224	2147483635
ecdsa-sha2-nistk233	SSH_PK_ECDSA_NIST_K233	2147483634
ecdsa-sha2-nistb233	SSH_PK_ECDSA_NIST_B233	2147483633
ecdsa-sha2-nistk283	SSH_PK_ECDSA_NIST_K283	2147483632
ecdsa-sha2-nistk409	SSH_PK_ECDSA_NIST_K409	2147483631
ecdsa-sha2-nistb409	SSH_PK_ECDSA_NIST_B409	2147483630
ecdsa-sha2-nistt571	SSH_PK_ECDSA_NIST_K571	2147483629

SSH Cipher	Secure Blackbox Encryption Algorithm	Priority
ecdsa-sha2-curve25519	SSH_PK_ECDSA_CURVE25519	2147483628
x509v3-ssh-rsa	SSH_PK_X509_SSH_RSA	2147483627
x509v3-ssh-dss	SSH_PK_X509_SSH_DSS	2147483626
x509v3-rsa2048-sha256	SSH_PK_X509_RSA2048_SHA256	2147483625
x509v3-ecdsa-sha2-nistp256	SSH_PK_X509_ECDSA_SHA2_NIST_P256	2147483624
x509v3-ecdsa-sha2-nistp384	SSH_PK_X509_ECDSA_SHA2_NIST_P384	2147483623
x509v3-ecdsa-sha2-nistp521	SSH_PK_X509_ECDSA_SHA2_NIST_P521	2147483622
x509v3-ecdsa-sha2-nistk163	SSH_PK_X509_ECDSA_SHA2_NIST_K163	2147483621
x509v3-ecdsa-sha2-nistp192	SSH_PK_X509_ECDSA_SHA2_NIST_P192	2147483620
x509v3-ecdsa-sha2-nistp224	SSH_PK_X509_ECDSA_SHA2_NIST_P224	2147483619
x509v3-ecdsa-sha2-nistk233	SSH_PK_X509_ECDSA_SHA2_NIST_K233	2147483618
x509v3-ecdsa-sha2-nistb233	SSH_PK_X509_ECDSA_SHA2_NIST_B233	2147483617
x509v3-ecdsa-sha2-nistk283	SSH_PK_X509_ECDSA_SHA2_NIST_K283	2147483616
x509v3-ecdsa-sha2-nistk409	SSH_PK_X509_ECDSA_SHA2_NIST_K409	2147483615
x509v3-ecdsa-sha2-nistb409	SSH_PK_X509_ECDSA_SHA2_NIST_B409	2147483614
x509v3-ecdsa-sha2-nistt571	SSH_PK_X509_ECDSA_SHA2_NIST_K571	2147483613
x509v3-ecdsa-sha2-curve25519	SSH_PK_X509_ECDSA_SHA2_CURVE25519	2147483612
ssh-ed25519	SSH_PK_ED25519	2147483611
ssh-ed448	SSH_PK_ED448	2147483610
rsa-sha2-256	SSH_PK_RSA_SHA256	2147483609
rsa-sha2-512	SSH_PK_RSA_SHA512	2147483608
ssh-dss	SSH_PK_DSS	2147483607
SSH_PK_RSA	SSH_PK_RSA	2147483606

Reports Overview

The reporting interface comes with a set of standard reports. These reports include a variety of 2D and 3D charting and graphing components and a full grid of data. Some of the reports are purely data detailed and have no charts. You can also create your own reports based on any Secret Server data, such as user, audit, permissions, and folders. You can create report categories to aid in the organization of your reports. Reports can be arranged to provide access to auditors and meet compliance requirements. These reports can be accessed in the **General** tab on the **Reports** page.

> Reports using SQL queries are displayed in UTC.

The *Security Hardening Report* checks aspects of Secret Server to ensure security best practices are being implemented. While Secret Server runs with all the items failing, administrators should be aware of possible security issues within an installation. For details on this, see "Report Page" on page 890.

The User Audit Report shows all secrets accessed by a user during a specified period.

Built-in Reports

Secret Server includes many pre-configured reports that you can run or use as templates for creating custom reports. Below are the reports shipped with current release of Secret Server:

> Unless otherwise designated, reports listed are available in all editions. However, older releases may not include all reports listed here.

Activity

- Active Secret Sessions
- Active Secret Sessions Count
- Custom Report Activity
- Database Configuration Audit
- Distributed Engine Activity
- Dual Control Audit
- Engine Status
- Event Subscription Activity
- Folder Activity
- Heartbeat Status
- Heartbeat Status by Day
- Internal Communication Changes
- IP Address Range Audit
- License Audit

- RPC by Day
- Secret Activity
- Secret Activity Today
- Secret Activity Yesterday
- Secret Template Activity
- Session Recording Errors
- Unlimited Administrator behavior
- Users Activity

Discovery Scan

- Discovery Scan Status
- GCP Discovery: What Instances do Service Accounts have access to?
- What computers have been successfully scanned?
- What computers in Active Directory no longer exist?
- What computers that exist have not been successfully scanned?
- What Secrets are pending import by Discovery?
- What Secrets failed to import by Discovery?

Folders

- What folder permissions exist for groups?
- What folder permissions exist?
- What folders can a user see?
- What folders can all users see?

Groups

- Group Membership
- Group Membership By Group

Legacy Reports

- Secret Expiration Health
- Secret Server Usage
- Secret Template Distribution
- Top Ten Viewers

Password Compliance

- Secret Password Compliance Statuses
- What Secrets Do Not Meet Password Requirements?

Report Schedules

- Report Schedules

Roles and Permissions

- What role assignments exist?
- What role permission assignments exist?
- What role permissions does a user have?

Secret Policy

- What Folders have Policies assigned?
- What Secrets have different Policies than their folders?
- What Secrets have policies assigned?

Secrets

- Secret Count per Site
- Secret Dependency Failures
- Secret Dependency Not Run
- Secret Dependency Overview
- Secret Dependency Status
- Secret Permissions Mismatch.
- Secret Templates without an expiration field?
- Secrets Failing Heartbeat
- Secrets Pending Heartbeat
- Secrets with Failed Password Change
- What file types have been uploaded to Secrets?
- What file types have been uploaded to Secrets? (Pie Chart)
- What Secret permissions exist for a group?
- What Secret permissions exist for a user?
- What Secret permissions exist?
- What Secrets are expiring this week?
- What Secrets can a user see?

- What Secrets can all users see?
- What Secrets changed passwords in the last 90 days?
- What Secrets Do Not Have Distributed Engines?
- What Secrets don't require approval?
- What Secrets have been accessed by a user?
- What Secrets have been accessed by an impersonated user?
- What Secrets have been accessed?
- What Secrets Have Distributed Engines?
- What Secrets have Expiration?
- What Secrets have failed Heartbeat?
- What Secrets have not changed passwords for over 90 days?
- What Secrets require approval?
- What Secrets require Comments?
- What SSH Command Menus do Secrets have?

System Reports

- Folder Permissions Report
- Folder Secrets Report
- Group Lookup Report
- Permission Lookup Report
- Privileged Behavior Analytics Configuration Activity
- Role Permissions Report
- User Access Report

User

- Active Users Custom Report
- Failed login attempts
- Secret Template Permissions by User
- Session resumed
- What SSH Command Menus do users have access to?
- What users have had an admin reset their password?
- Who hasn't logged in within the last 90 days?

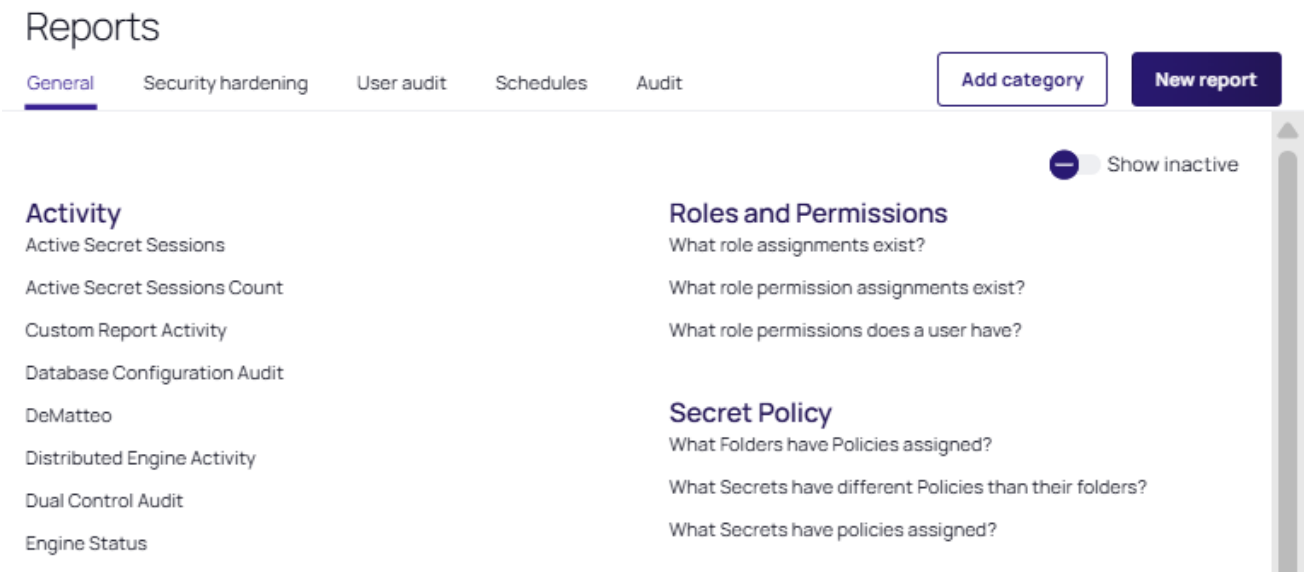
Creating and Editing Reports

Secret Server provides robust reporting capabilities to help you analyze and manage your privileged access data effectively. With customizable reports, you can gain insights into user activity, security compliance, and system performance.

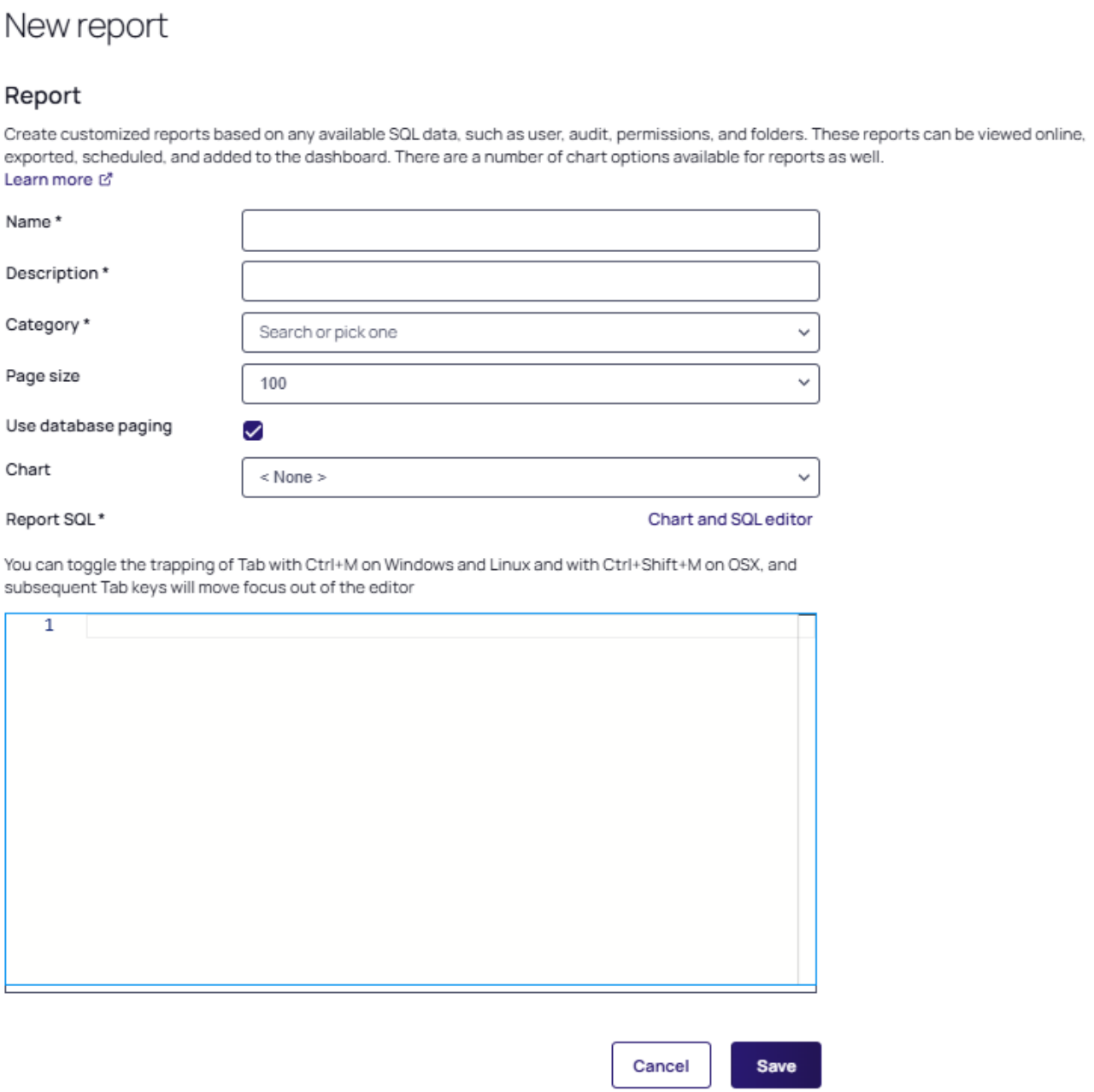
This guide walks you through the process of creating, modifying, and customizing reports in Secret Server. Whether you need to generate pre-built reports or tailor them to specific security and compliance requirements, you'll find the tools and options necessary to streamline reporting for your organization.

Creating a Custom Report

1. Click **Reports** on the main menu. The **Reports** page appears:



2. Click the **New Report** button. The **New Report** page appears:



3. Type the report name in the **Name** text box. This is the name that is displayed on the Reports page as a link underneath its containing category.
4. Type a description in the **Description** text box. This is displayed in the View Report tab of a report. It is also used as the Tooltip when hovering over the report name on the Reports page.
5. Click the **Category** dropdown list to select the category the report will appear in on the Reports page.
6. Click the **Page Size** dropdown list to select the page size limit for the data displayed in the grid.
7. Select the **Use Database Paging** checkbox if desired. See Database Paging for details.

8. Click the **Chart** dropdown list to select the type of chart to use for displaying results. If set to <None>, a grid displays.
9. Paste your script in the **Report SQL** text box. See Report SQL Scripts for details.
10. Click the **Save** button. The new report's page appears and now appears on the reports page.

Editing Reports

To edit a report:

1. Click the **Reports** menu item. The Reports page appears, listing all reports.
2. Click the name of the report you wish to edit, which is a link. The report's page appears.
3. Click the **Edit** button. The Edit Report page appears. See Creating a Custom Report for details about the parameters.

> The SQL script text cannot be edited for (non-custom) system reports.

Report SQL Scripts

Overview

The best way to create SQL scripts is to view existing ones and the Secret Server database structure. Click any existing report link from the **Reports** page and select the **Edit** button. The SQL script appears in the **Report SQL** text box.

> You cannot edit (non-custom) system reports. You can view their parameters in the Edit page, including their SQL script, which you can copy for use, but no modifications can be made or saved.

Dynamic Parameters

Reports support the embedding of certain parameters into the SQL so you can dynamically change the resulting data set. Another option available for custom reports is to apply a different color to returned rows dependent on certain conditions. For more information as well as examples, see "Using Dynamic Parameters in Reports" on page 899 for details.

Viewing Secret Server SQL Database Information

You can show the SQL database information of Secret Server to assist with creating custom reports.

When "Creating a Custom Report" on page 887, in the **New Report** page select the **Chart and SQL Editor** link. **The Report SQL** page appears and it is blank. Click **Help** and search for specific tables with examples of data, data types and whether they are nullable. This data is available for search to assist in creating custom reports.

You can click the **Run SQL** button at the bottom of the page to see a preview of the chart generated by the report. The resulting chart displays in the Report Preview section at the bottom of the page.

Database Paging

Database paging allows the database to load large reports more quickly. We recommend database paging if the query is expected to pull large amounts of data for a report.

> Implementing database paging may not work if the SQL query uses certain keywords, including `TOP`, `OPTION`, `INSERT`, `UNION`, `WITH`, or aliases containing the word `FROM`.

Example queries:

- Successful query: `SELECT * FROM tbSecret WHERE NAME LIKE 'Test%'`.
- Failing query: `SELECT TOP 10 * FROM tbSecret WHERE SecretName LIKE 'Test%'`.

Deleting or Undeleting Reports

To delete or undelete a report.

- **Delete:** To delete a report, click the **Delete** button.
- **Undelete:** To undelete a report, you must navigate to the Reports Edit page as deleted reports are not visible on the Reports View page. On the Reports Edit page, click the **Show Deleted** button. This displays a Deleted Report category, which contains all the deleted reports. Either drag the report to a report category that is not deleted or click the report name to go into its Report View page. In there, click the **Undelete** button.

Modifying Report Categories

For details on the Show Deleted button, see "Deleting or Undeleting Reports" above.

- **Rearrange:** Any item with the icon can be dragged and dropped to a new location. Report categories can be moved anywhere within the page. Reports can be moved from one report category to another.
- **Create New:** Click **Create Report Category** and specify a category name and description on the following page. Note that the Report Category Description is used as the tooltip for the report category on the Reports View page.
- **Delete:** Click the icon next to the report category name. This deletes all the reports in the category. To undelete the reports, see "Deleting or Undeleting Reports" above.
- **Edit:** Click the icon next to the report category name to change the name or description of the category.

Report Page

Reports General Tab

See "Built-in Reports" on page 883 for the most up-to-date list of reports included.

The reports are listed under the report categories. To view a report, click on its name. This takes you to the **Report View** page.

You can view a record of all the actions performed on reports by clicking on the **View Audit** button.

For details on the **Edit** button, see "Creating and Editing Reports" on page 887.

The **Create it** link is a shortcut for creating a new report.

You can adjust the look of the Reports View page. The report categories as well as the reports can be rearranged on the page. To do this, click **Edit** on the Reports page.

Reports Security Hardening Tab

The Security Hardening Tab configures aspects of Secret Server to ensure security best practices are being implemented. While Secret Server runs with all the items failing, administrators should be aware of possible security issues within an installation. Below is an explanation of the different features:

Configuration Section

- **Allow Approval for Access from Email:** This is a convenience option that allows users to approve or deny a secret access request by clicking a link in the request email sent by Secret Server. Allow Approval From Email does not require a user to authenticate with Secret Server when approving access to a secret. This can be a security concern if the approver's email account becomes compromised. Turn Allow Approval From Email off to get a pass result. Secret Server.
- **Browser AutoComplete:** Browser AutoComplete allows Web browsers to save the login credentials for the Secret Server login screen. These credentials are often kept by the Web browser in an insecure manner on the user's workstation. Allowing AutoComplete also interferes with the security policy of your Secret Server by not requiring the user to re-enter their login credentials on your desired schedule. To prevent the AutoComplete feature, disable the Allow AutoComplete option on the Configuration page.
- This security check will fail if the file attachment restrictions is not enabled. This check will return warnings if a potentially dangerous file extension is allowed, maximum file size is not specified, or maximum file size is greater than 30 MB.
- **Force Password Masking:** Password masking prevents over-the-shoulder viewing of your passwords by a casual observer (when masked, passwords show as *). To activate this option, click to select the **Force Password Masking** option on the **Configuration** page.
- **Frame Blocking:** Frame blocking prevents the Secret Server site from being placed in an iFrame. This is to prevent clickjacking attacks. There may be legitimate reasons for placing Secret Server in a frame, such as embedding the UI in another site. To turn frame blocking on, enable the setting under the Security tab in Configuration.
- **Login Password Requirements:** Login passwords can be strengthened by requiring a minimum length and the use of various character sets. A minimum password length of 8 characters or longer is recommended. In addition, all character sets (lowercase, uppercase, numbers and symbols) are required to get a pass result. Turn on these login password settings on the Configuration page.
- **Maximum Login Failures:** The maximum number of login failures is the number of attempts that can be made to login to Secret Server as a user before that user's account is locked. A user with user administration permissions is then required to unlock the user's account. The maximum failures allowed should be set to 5 or less to get a pass result. Change the "Maximum Login Failures" settings on the Configuration page. Secret Server
- **Remember Me:** Remember Me is a convenience option that allows users to remain logged in for up to a specific period. This setting can be a security concern as it does not require re-entry of credentials to gain access to SS.

Turn Remember Me off on the Configuration page to get a pass result. It must be set to be valid for 1 day or less to not get a fail result.

- **Secure Session and Forms Auth Cookies:** Cookies contain potentially sensitive information that can allow users to log onto application. By default, cookies are not marked with the secure attribute. That is, **they are transmitted unencrypted when a user accesses Secret Server through HTTP instead of HTTPS**.

  For more information about how to secure your cookies, see Secure ASP Session and Forms Authentication Cookies (KB).

- **Web Service HTTP Gets Allowed:** Web service HTTP get requests are allowed. Allowing HTTP GET requests allows REST-style calls to many Secret Server Web service methods. This can be a security concern because simply clicking a link to the Web service, created by a malicious user, would cause it to be executed.
- **Zero Information Disclosure Error Message:** Replace all error messages with a custom "contact your admin" message. Error messages can be very helpful when diagnosing installation and configuration issues. However, having errors displayed to a potential attacker can provide him or her with the critical information they need to perform a successful attack.

File Attachment Restrictions

File attachment restrictions allows administrators to configure what kind of file attachments can be uploaded to secrets. This helps protect users from being tricked into downloading a malicious secret attachment. The file extension and maximum file size can be specified, such as:

```
*.7z, *.bmp, *.ca-bundle, *.cer, *.config, *.crt, *.csr, *.csv, *.dat, *.doc, *.docx, *.gif,
*.gz, *.id-rsa, *.jpeg, *.jpg, *.json, *.key, *.lic, *.p7b, *.pcf, *.pdf, *.pem, *.pfx,
*.pkey, *.png, *.ppk, *.pub, *.tar, *.tif, *.tiff, *.tpm, *.txt, *.vdx, *.vsd, *.vsdx, *.xls,
*.xlsx, *.xml, *.zip
```

This security check will fail if the file attachment restrictions is not enabled. This check will return warnings if a potentially dangerous file extension is allowed, maximum file size is not specified, or maximum file size is greater than 30 MB.

Database Section

- **SQL Account Using Least Permissions:** Use the fewest Secret Server permissions as possible in the SQL Account used to access the database. We recommend using a least permission approach where the account only has dbOwner. See "SQL Server 2016 Standard Edition Installation" on page 111.
- **SQL Server Authentication Password Strength:** SQL Server authentication requires a username and password. The password must be a strong password to get a pass result. Strong passwords are 8 characters or longer and contain lowercase and uppercase letters, numbers and symbols. The SQL Server authentication credentials in use can be changed by going to the installer (installer.aspx) and changing them on Step 3. A pass result is also given if Windows authentication is used to authenticate to SQL Server.
- **SQL Server Authentication Username:** The SQL Server authentication username should not be obvious. The use of "sa", "Secret Server" or "secretserver" triggers a fail result. The SQL Server authentication credentials in use can be changed by going to the installer (installer.aspx) and changing them on Step 3. A pass result is also given if Windows authentication is used to authenticate to SQL Server.

- **Windows Authentication to Database:** Windows authentication takes advantage of Windows security to provide secure authentication to SQL Server. The SQL Server authentication options can be changed by going to the installer (installer.aspx) and changing them on Step 3. Please see "Installation" on page 65 for instructions on configuring Windows authentication to SQL Server.

Environment Section

- **Application Pool Identity:** The Application Pool identity `GAMMA\ss_iis_svc` appears to be a member of the administrators group on the system. This puts the system at risk by giving more access than necessary.
- **DPAPI or HSM Encryption of Encryption Key:** Encrypt your Secret Server encryption key, and limit decryption to that same server. Data Protection API (DPAPI) is an encryption library that is built into Windows operating systems. It allows encryption of data and configuration files based on the machine key. Enabling DPAPI Encryption in Secret Server protects the Secret Server encryption key by using DPAPI, so even getting access to the Secret Server encryption key is not enough to be useful–the machine key is required. If you enable this option, back up your encryption key first, as a DPAPI encrypted file can only be used by the machine it was encrypted on.

SSL Section

> SSL needs to be running with at least a 128-bit key size to get a pass result. A warning result indicates your key size is less than 128 bits. A fail result indicates you are not using SSL.

- **Require SMTP SSL:** SMTP SSL is required to ensure that all communication between Secret Server and the email server is encrypted. Enable the "Use SSL" option in Secret Server to get a pass result.
- **Require SSL:** Secure Sockets Layer (SSL) is required to ensure that all communication between the Web browser and Secret Server is encrypted and secure. Once the SSL certificate is installed, Force HTTPS/SSL in Configuration to get a pass result. Please see the "Installing Self-Signed SSL Certificates" on page 430 Knowledge Base article for instructions.
- **SSL/TLS Hash:** Check the digest algorithm of the certificate. If the algorithm is SHA1, this check returns a warning because SHA1 is being phased out. If the digest algorithm is MD2, MD4, or MD5, the check will fail because they are not secure. SHA256, SHA384, and SHA512 will pass. This check fails if Secret Server cannot be loaded over HTTPS.
- **SSL/TLS Key:** Check the key size of the HTTPS certificate used. If it is RSA or DSA, the key must be at least 2048-bit to pass. If the signature algorithm of the certificate is ECDSA, the key size must be at least 256-bit to pass. If the algorithm of the certificate is unknown, the result shows "unknown. This check fails if Secret Server cannot be loaded over HTTPS.
- **SSL/TLS Protocols:** Check for legacy SSL or TLS protocols, which should not be used in a secure environment. If the server accepts SSLv2 or SSLv3 connections, this check will fail. SSLv2 is not considered secure for data transport, and SSLv3 is vulnerable to the POODLE attack. If this server does not support TLSv1.1 or TLSv1.2, this check will give a warning because they are recommended. The SSL certificate used may affect what protocols can be used, even if they are enabled. This check will fail if Secret Server cannot be loaded over HTTPS.
- **Using HTTP Strict Transport Security:** HTTP Strict Transport Security (HSTS) is an additional security layer for SSL. HSTS allows SS, Password Reset Server, or Group Management Server to inform browsers that it should

only be accessible over HTTPS. With this setting enabled, visitors are automatically are redirected by their browser to the HTTPS-enabled site.

Reports User Audit Tab

User Audit Reports show all secrets accessed by a user during a specified period. For a more detailed explanation, see "Built-in Reports" on page 883.

Reporting and Dual Controls

If there are requirements around protecting potentially personally identifying information when running reports or viewing recorded sessions, you can enforce that another user has authorized you by enabling dual control for a secret or Report. When enabled a user in the approver group must enter in their credentials before a report or session can be viewed.

Once the approver has entered their credentials, the resource can be accessed.

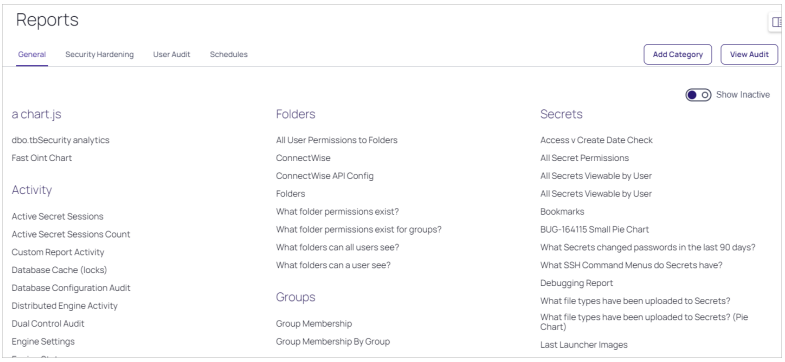
To configure dual control, perform the following steps:

1. Click **Administration** > **Diagnostics, Logs, Security** > **Dual Controls**.
2. On the Dual Controls page, click the **Create New** button.
3. On the **Create New** page in the **Type** drop-down box, select the report type you wish to modify.
4. In the drop-down list below **Add**, click the box next to each party you'd like to add as a Valid Approver. Their identities will automatically appear in the **Valid Approvers** field.

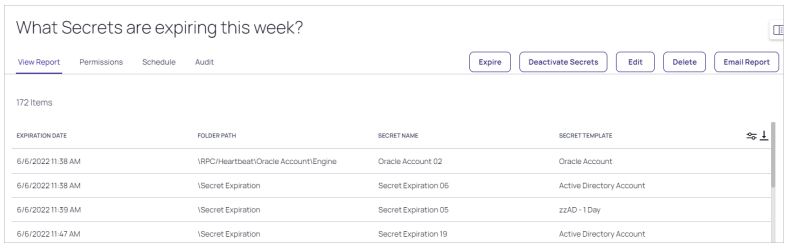
   When you are finished, click the **Create New** button. The Dual Control page appears, displaying the parameters you have set.

Saving Reports to File

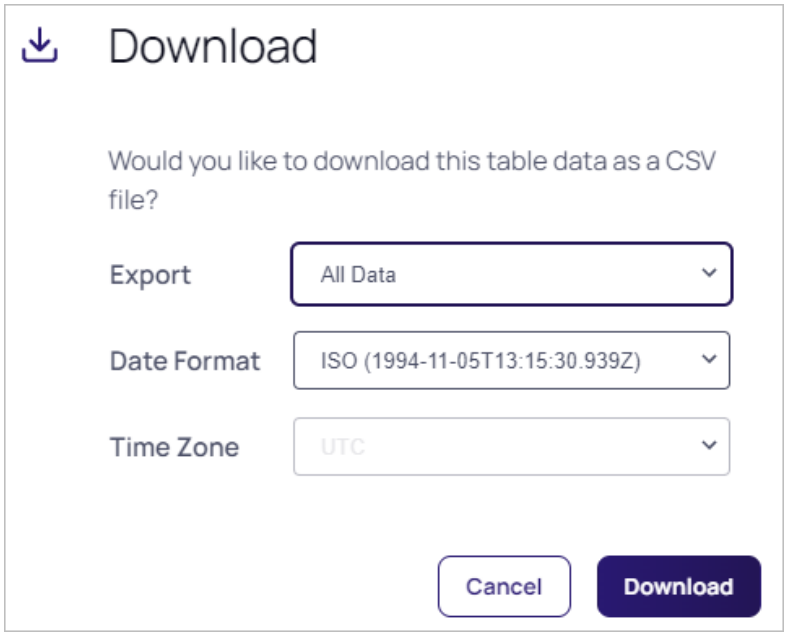
1. Click the **Reports** menu item. The Reports page appears:



2. Click the link for the desired report. Its page appears:



3. Click the  button in the top right of the item table. The download popup appears:

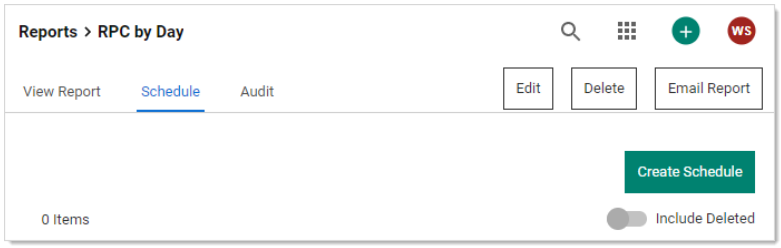


4. Click the **Export** dropdown list to choose which data to download.
5. Click the **Date Format** dropdown list to choose the date format.
6. Click the **Time Zone** dropdown list to choose the time zone.
7. Click the **Download** button. The reports save as a CSV file to your browser's default download location.

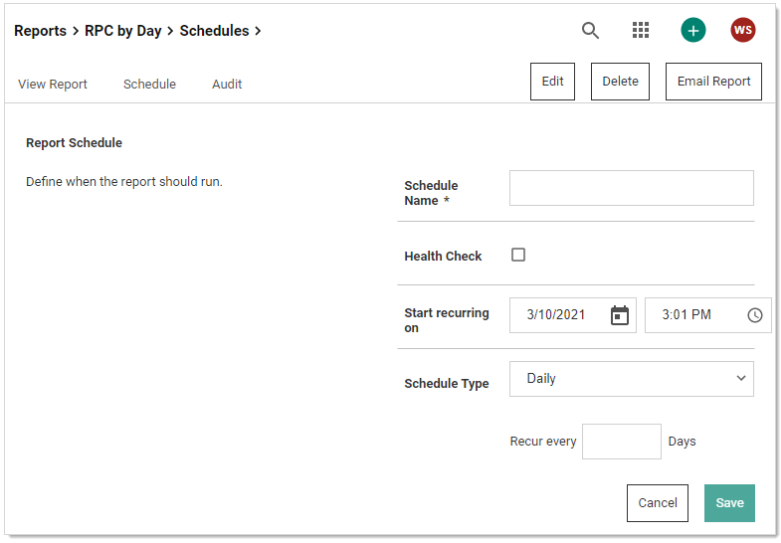
Scheduled Reports

Creating New Schedules for Reports

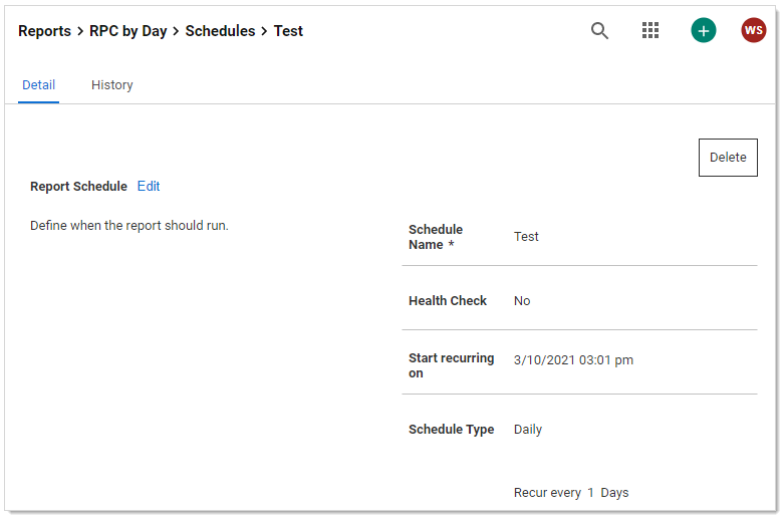
1. To create a schedule for a report, click the **Schedules** tab on the **Report View** page:



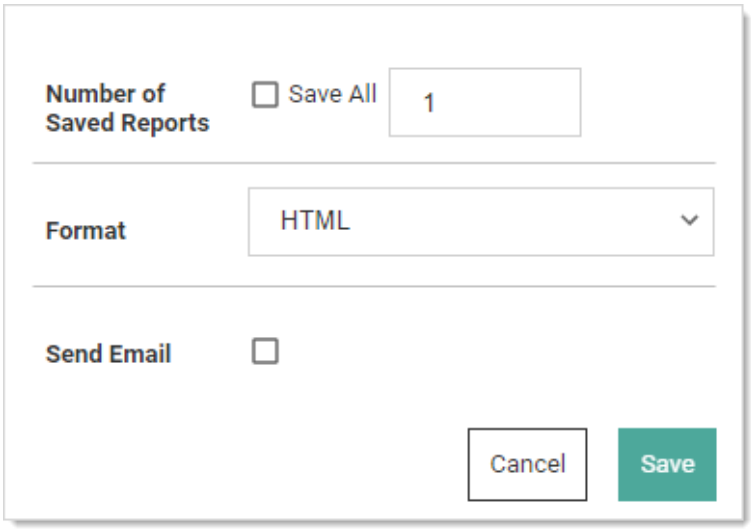
2. Click the **Create Schedule** button. The Report Schedule page appears:



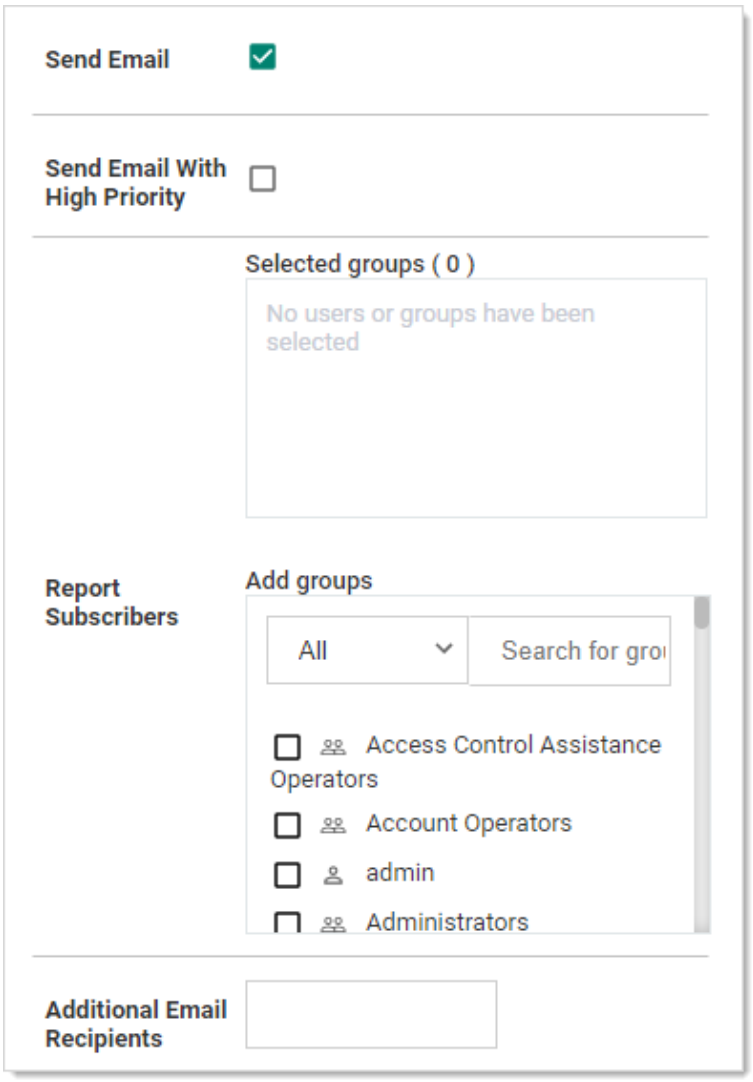
3. Configure the report settings as listed in Editing Schedule Settings
4. Click the **Save** button. The page for the new report schedule appears:



5. The report will now be saved for you, saving only one report at a time. If you want more saved or want it emailed to you according the schedule:
   a. Click the **Edit** link in the **Report Distribution** section. The section becomes editable:



   b. Either click to select the **Save All** check box or type a new number in the **Number of Saved Reports** text box. Remember, saving reports can use a lot of disk space.
   c. Click the **Format** dropdown list to select the report format:
      - HTML: Save the report as an HTML file.
      - CSV: Save the report as a comma separated value file for importation into a spreadsheet.
   a. Click to select the **Send Email** check box to have the report emailed to you at the reporting interval. An email section appears:



b. Click the **All** dropdown list in the **Report Subscribers** section to choose the domain to look for users and groups.

c. Click the check boxes next to the users or groups you want to send the report to in the **Report Subscribers** section. You can also search for the same in the provided search box at the top of the section. The users or groups appear in the Selected Groups text box.

d. Type any additional email addresses in the **Additional Email Recipients** text box.

e. Click the **Save** button.

Viewing Existing Report Schedules

1. To view existing schedules for a report, click **Schedule** on the Report View screen. A list of existing schedules for the report appear in the grid.
2. To view the details of a schedule, click the schedule name in the grid.

3. (Optional) Deleted schedules can be made visible by checking the **Show Deleted** box at the bottom of the grid.
4. Click the **View** link in the History column of the grid to view the history of all generated reports for that schedule.

Editing Schedule Settings

When viewing a report, click Schedule and then the name of the report schedule to modify it. The following configuration options are available:

- **Schedule Name:** This is the name of the schedule for the report. This name must be unique to the Secret Server installation.
- **Health Check:** This sends an email notification only when the report contains data.
- **Recurrence Schedule:** This specifies the schedule runs every X number of days, weeks, or months, with the option to specify days of the week or month as well. The date and time that the report schedule is effective can be specified in this section as well.
- **Save Generated Reports:** This saves the history of generated reports in the database for later viewing. Enabling this setting also allows you to specify the number of generated reports to save.
- **Send Email:**Secret Server sends an email containing the generated report every time the schedule runs. Enabling this setting also allows you to specify whether the email is sent with the high priority flag and a list of Secret Server users or groups that receive the generated report email. Add additional email recipients in the text box below the subscribers, separating recipients with a semi-colon.

The following configuration options appear if the report being scheduled contains at least one dynamic parameter in the SQL of the report:

- **User Parameter Value:** Value of the `#USER` parameter to set in the report when it is generated.
- **Group Parameter Value:** Value of the `#GROUP` parameter to set in the report when it is generated.
- **Start Date Parameter Value:** Value of the `#STARTDATE` parameter to set in the report when it is generated.
- **End Date Parameter Value:** Value of the `#ENDDATE` parameter to set in the report when it is generated.

Using Dynamic Parameters in Reports

As version 7.0, Secret Server allows creation of Reports using custom SQL.

Reporting supports embedding certain parameters into the SQL to give the viewer controls to dynamically change the report. The supported parameters are:

Primary Parameters

#STARTDATE

This displays a calendar picker on view and returns a date. This defaults to beginning of the year and truncates the hours and minutes to 12:00 AM.

Example: display all users who have logged in after a certain date:

```
SELECT
  Domain,
  Username,
  LastLogin
FROM tbUser
  LEFT JOIN tbDomain ON tbUser.DomainId = tbDomain.DomainId
WHERE
  LastLogin > #STARTDATE
```

#ENDDATE

This displays a calendar picker on view and returns a date. This defaults the current day and truncates the hours and minutes to 11:59 PM.

Example: display all users who have logged on a certain date:

```
SELECT
  Domain,
  Username,
  LastLogin
FROM tbUser
  LEFT JOIN tbDomain ON tbUser.DomainId = tbDomain.DomainId
WHERE
  LastLogin > #STARTDATE
AND
  LastLogin < #ENDDATE
```

#USER

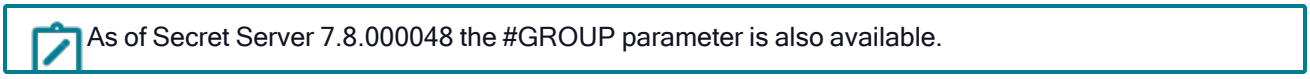
This displays a user dropdown list with all active users on view and returns an user id. This defaults to the current logged in user.

Example: display all audit entries for a certain user:

```
SELECT
  tau.UserIdAffected,
  tau.[Action],
  tau.Notes,
  tau.DateRecorded,
  tau.IpAddress,
  tau.MachineName,
  tau.DatabaseName,
  tu.UserId,
  tu.UserName
FROM tbAuditUser tau INNER JOIN tbUser tu ON tau.UserId=tu.UserId WHERE tu.UserId=#USER
```

#ORGANIZATION

This is an internal parameter that returns the current instance's organization code. This is only useful for Secret Server Online (a legacy product, which is *not* the same as Secret Server Cloud). Do not use this parameter in your reports for either Secret Server On-Premises or Secret Server Cloud.



#GROUP

Displays a group dropdown list with all active groups on view and returns a group id. This defaults to the All Vault Users group.

Example: display the group details of the selected group:

```
SELECT
  GroupID,
  GroupName,
  Active
FROM tbGroup
WHERE GroupID = #GROUP
```

#FOLDERID

Displays a folder picker that shows all Folders and returns a folder id.

Example: Display secret names in a selected folder:

```
SELECT
  s.SecretName
FROM tbSecret s
WHERE s.Folderid = #FOLDERID
```

#FOLDERPATH

Displays a folder picker that shows all folders and returns the path of the folder.

Example: display folders that are child folders of the selected path:

```
SELECT *
FROM tbFolder f
WHERE FolderPath LIKE '%' + #FOLDERPATH + '%'
```

#CUSTOMTEXT

Displays a text input where a user can put in arbitrary free text for searching.

Example: display secrets that have names that contain the text input:

```
SELECT *
FROM tbFolder f
WHERE FolderPath LIKE '%' + #CUSTOMTEXT + '%'
```

Additional Parameters

The following additional parameters can be used to make your report more dynamic:

Parameters

Table: Additional Parameters

Parameter Name	Description
#ENDCURRENTMONTH	The last day of current month
#ENDCURRENTYEAR	December 31st of the current year
#ENDPREVIOUSMONTH	The last day of the previous month at 11:59:59 PM
#ENDPREVIOUSYEAR	December 31st of the previous year
#ENDTODAY	End of today at 11:59:59 PM
#ENDWEEK	End of the current week (Sunday) at 11:59:59 PM
#ENDYESTERDAY	End of Yesterday at 11:59:59 PM
#STARTCURRENTMONTH	The first day of current month
#STARTCURRENTYEAR	January 1st of the current year
#STARTPREVIOUSMONTH	The first day of the previous month at 12:00 AM
#STARTPREVIOUSYEAR	January 1st of the previous year
#STARTTODAY	Beginning of today at 12:00 AM
#STARTWEEK	Beginning of the current week (Monday) at 12:00 AM
#STARTYESTERDAY	Beginning of yesterday at 12:00 AM

Example

For example, the following script would give you a list of all users who have logged on during the last calendar month:

```
SELECT
  Domain,
  Username,
  LastLogin
FROM tbUser
  LEFT  JOIN tbDomain ON tbUser.DomainId = tbDomain.DomainId
WHERE
  LastLogin BETWEEN #STARTPREVIOUSMONTH AND #ENDPREVIOUSMONTH
```

As of Secret Server 7.8.000048, the #STARTWEEK and #ENDWEEK parameters are available.

Coloring Your Reports

Another option when creating reports is to include a Column in your SQL query called "Color" this will give the row that particular color. See HTML Color Names.

For example, to show users who haven't logged in within 90 days in Red:

```
SELECT DisplayName
  CASE
    WHEN LastLogin < GetDate() - 90 THEN 'Red'
    ELSE 'White'
  END AS Color
FROM tbUser
```

Viewing Auditing for a Report

To view auditing for all actions performed on a report, do the following:

1. Click Reports from the left navigation panel.
2. Click the report for which you wish to view an audit.
3. At the far right end of the page, click the button labeled **View Audit**. The audit information for the record appears in a pop-up message. For more information, See "Built-in Reports" on page 883.

Viewing Reports

On this page you see the graph, chart, grid, and more for the report. To see a grid representation of the report, click the **Show Data** link to expand that area. If there is no data, then no graph is visible and the text "There are no items" displays in the Show Data section.

Some reports use dynamic values like user, start date, and end date. Adjust these values to generate the report you need. Click the **Update Report** button to generate the new report.

The **Edit** button allows you to alter the report to fit your requirements. See the Creating and Editing a Report topic for details.

RPC, Heartbeat, and Key Rotation

In Secret Server, Remote Password Changing (RPC) allows for the automatic updating of remote account passwords when a secret expires, ensuring synchronization with domain password policies and enhancing security by generating strong passwords ("RPC Overview" below). Heartbeat functionality verifies the validity of stored credentials by attempting to authenticate with the target system, flagging any discrepancies to prevent out-of-sync credentials ("Heartbeat Overview" on page 1042). Key rotation, including SSH key rotation, involves regenerating and updating public/private key pairs and their passphrases, ensuring that keys remain secure and compliant with organizational policies.

RPC Overview

Remote Password Changing (RPC) allows secrets to automatically update a corresponding remote account. You can set secrets for automatic expiration, followed by automatic strong password generation and a remote password update to keep the subject accounts synchronized with Secret Server.

RPC allows Secret Server to rotate passwords to meet domain password policy requirements. In most cases, RPC is configured with the secret "auto change" setting set to true. This causes the secret to rotate the password as soon as it expires. The "auto change schedule" setting changes the password on a set schedule, rather than when it expires. This provides the ability to change passwords when network activity is lower. You have a choice of changing the password as soon as the schedule interval arrives or only after the secret expires *and* the interval arrives. It is important to choose a large enough interval to complete all your password changes, otherwise any excess changes will have to wait for the next interval. Because the smallest interval is one day, this is only relevant if you have thousands of changes. If Secret Server fails to change a remote password, an alert states there are secrets out of sync.

You can pair secrets with Secret Server checkout, which is Delinea's one-time password functionality (not the same as "TOTP" on page 442). This allows you to rotate the password on a particular expiration schedule and limit the password to a single user for a set time period, after which it is changed. This is for situations where you need the password to change after every use, such as vendors who need temporary access to a server or system. For additional security on sensitive systems, approval workflow or session recording can be paired with checkout to add layers of authentication to gain access to the secret and track how that secret is used.

Regardless of the timing of password change, you may want to rotate dependent resources (dependencies) right after you rotate the password on a secret. For example, a Windows domain account could be a service account that starts many windows services. In the event that you rotate that password, you would need to also rotate the password for this account on the services which start using that account. If you do not, starting those services will fail the next time they are restarted, which could cause other components to fail too. You can create dependencies on a secret for scheduled tasks, application pools, or services (with or without using PowerShell to undertake tasks at rotation time).

We have a large number of out-of-the-box RPC changers, which are expandable by writing your own SSH, SQL or PowerShell scripts to do RPC, which can get information from the secret. See "Secret Dependencies for RPC" on page 934 and the "Password Changer List" on page 909.

Custom Password Changers

The Password Changers Configuration page can be accessed by navigating to **Admin > Remote Password Changing > Configure Password Changers**.

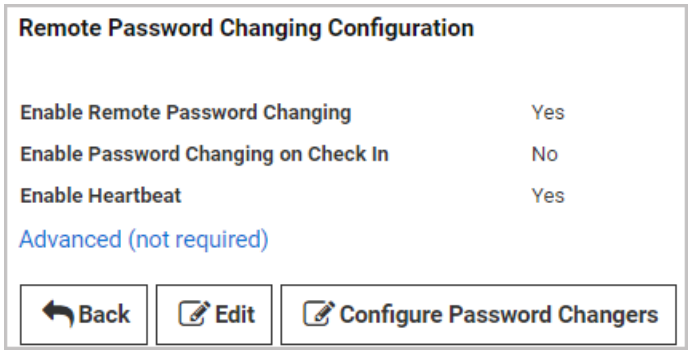
There are a few password changing types that allow the user to enter in specific commands that are sent to the computer where the password is changing. This enables the system to accommodate for differences in the standard password change procedure. For example: The Unix system that is being changed prompts for the current password twice instead of only once before asking for the new password.

Creating a Custom Password Changer

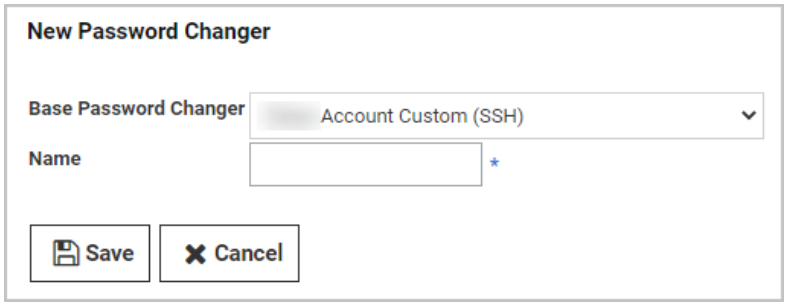
To create a custom password changer, follow the procedure below.

> Before creating the password changer and attempting to change a password through Secret Server, we recommend that you test the command set you are using directly.

1. From the **Admin** menu, select **Remote Password Changing**.
2. Click **Configure Password Changers**, then scroll to the bottom of the page and click **+New**.



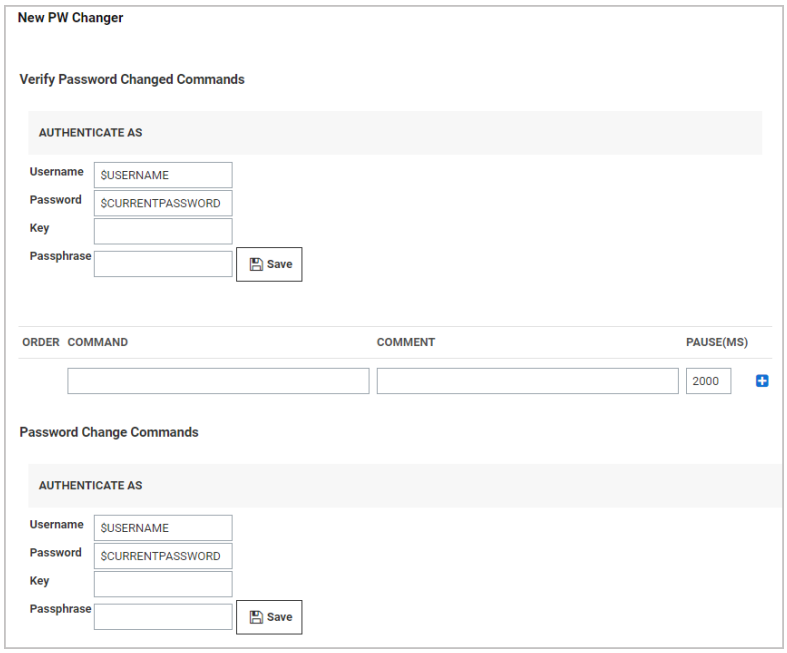
3. Choose a **Base Password Changer** with a command set that most closely matches the type of password changer you are creating, as this determines which customizable parameters and test actions are available to you. To create a custom SSH password changer, choose a base password changer with a name that ends in `(SSH)`.



4. Give your new password changer a **Name** and click **Save**.
5. Edit the **Password Change Commands** to contain the command set you need.
   - Use the Delete button to remove a row.
   - Use the Edit button to modify a row.

- Use the up and down arrows to move a row.

6. Use the Plus button to save a row.

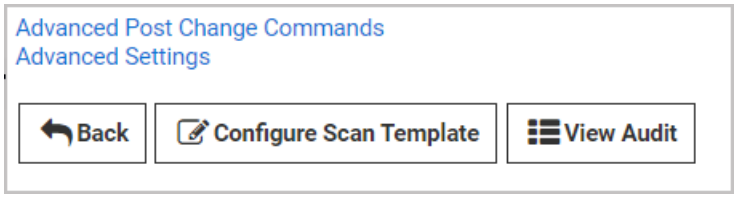


7. Edit the **Verify Password Changed Commands** to create the command set for checking that the password is valid. These commands are used by heartbeat and after a password change to verify that the change was successful.
8. When you are finished editing the commands, scroll to the bottom and click **Back** to return to the overview screen and access test actions for your new password changer. To edit advanced commands and settings, see the instructions below.

Advanced Post Change Commands

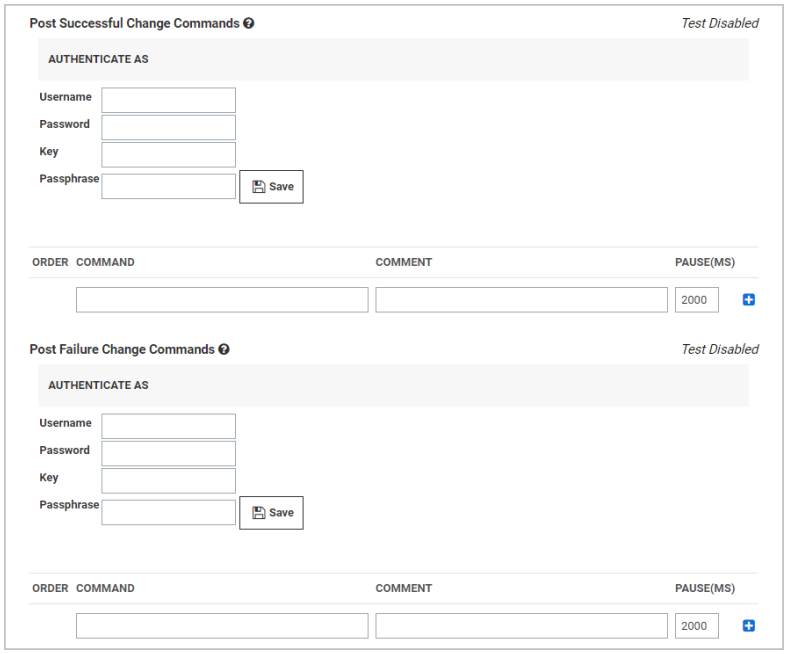
To modify advanced post change commands, do the following:

1. Scroll to the bottom of the page and click **Advanced Post Change Commands**.



2. Change the commands as desired in the under **Post Successful Change Commands** and **Post Failure**

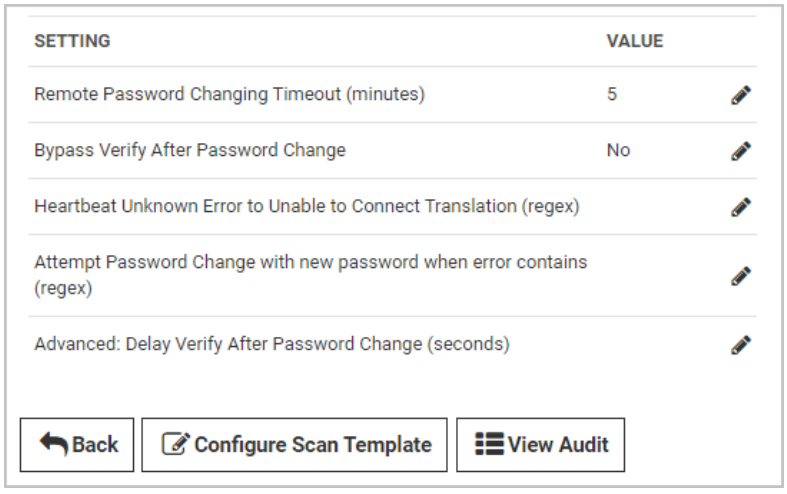
Change Commands.



Advanced Settings

To modify advanced settings, do the following:

1. Scroll to the bottom of the page and click **Advanced Settings**.
2. Change the settings as desired in the under **Post Successful Change Commands** and **Post Failure Change Commands**.



Before attempting to change a password through Secret Server using your new custom password changer, we recommend that you test the command set you are using.

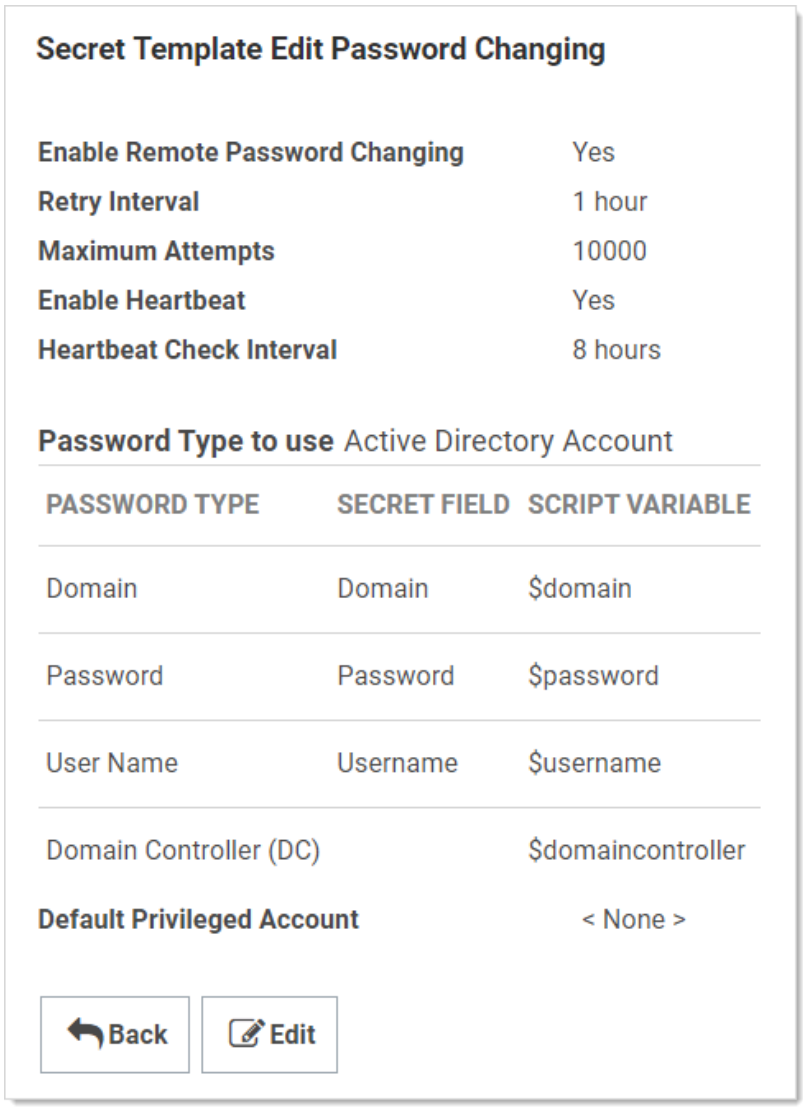
A Note About Commands

Any term in these commands preceded by `$` will reference a secret template field. Any term preceded by `$[1]$` refers to the Secret template field of a linked Secret. If you need to reference a secret template field, make sure you are using the exact secret template field name.

To use your new password changer, you will need to assign it to a secret template. See "Assigning a Password Changer to a Secret Template" on page 916.

Mapping Account Fields for Custom Templates Using RPC

All the secret templates with the prefix RPC have RPC configured by default. For creating a custom template that uses RPC it can be configured from the Secret Template Designer. **Enable Remote Password Changing** must be turned on for secrets created from the template to make use of this feature. Select the password type for the account and map the text-entry fields to be used for authenticating to the remote server. The secret fields must be mapped to the corresponding required text-entry fields based on the password change type. Do that in the **Secret Template Edit Password Changing** page for the secret template:



The **Retry Interval** text box is the amount of time that a secret waits before once again attempting to change a password after a password change is unable to succeed.

The **Default Privileged Account** text box is the secret that is set as the privileged account for all new secrets that are created with this secret template. Changing this does not affect any existing secrets.

General Information

Privileged Accounts and Reset Secrets and Password Changer List.

Password Changer List

Overview

Secret Server includes many pre-configured password changers that are used by Remote Password Changing (RPC). The following are commonly used password changers, and the list is always growing.

To see the latest list, go to Admin > RPC > Configure RPC.

Secret Server can use scripted password changers for devices that support SSH or Telnet (this allows for flexibility in changing passwords on less common devices). You can also run custom RPC PowerShell scripts to conduct password changes to other platforms.

The AD password changer has an RPC "timeout minutes" advanced setting. This setting only applies when using "Password Change By Admin Credentials."

List

The following are the current built-in password changers (click the link to see the supporting secret template's page, if available):

- "Microsoft AD Secret Template for RPC" on page 966
- Active Directory LDS
- AD LDS
- Amazon IAM Console Password Privileged Account
- Amazon IAM Key
- AS/400 IBM iSystem
- Azure AD
- Blue Coat Account Custom (SSH)
- Blue Coat Enable Password Custom (SSH)
- Cisco Account Custom (SSH)
- Cisco Account Custom (Telnet)
- Cisco Enable Secret Custom (SSH)
- Cisco Enable Secret Custom (Telnet)
- Dummy PowerShell Changer
- Entra ID
- ESX/ESXi (API)
- F5 BIG-IP Root Account (SSH)
- Generic Discovery-Only Credentials
- Generic ODBC (DataSource)
- Generic ODBC (PostgreSQL DataSource)
- Generic ODBC (SQL DataSource)
- Google IAM Service Account Key

- HP iLO Account Custom (SSH)
- IBM iSeries Mainframe
- Juniper Account Custom (SSH)
- LDAP (Active Directory)
- LDAP (DSEE)
- LDAP (OpenLDAP)
- MySQL Account
- New Cisco PW Changer
- New PW Changer
- "Okta Secret Template for RPC" on page 968
- Oracle Account
- Oracle Account (AS SYS)
- Oracle Account (DataSource)
- Oracle Account (TCPS)
- Oracle Account (Template Ver 2)
- PostgreSQL Account (x64)
- PowerShell Script **
- SAP Account**
- "ServiceNow Template for RPC" on page 974
- Snowflake Secret Template for RPC
- SonicWall NSA Web Admin Account
- SonicWall NSA Web Local User Account
- SQL Server Account
- SSH Key Rotation **
- SSH Key Rotation (No Password) **
- SSH Key Rotation Privileged Account **
- SSH Key Rotation Privileged Account (No Password) **
- Sybase Account
- Unix Account (SSH)
- Unix Account (Telnet)
- Unix Account Custom (SSH)
- Unix Account Custom (Telnet)
- Unix Account SU Takeover (SSH)

- Unix Account SUDO Takeover (SSH)
- Unix Root Account Custom (SSH)
- WatchGuard Custom (SSH)
- Web User Account (built-in support for AWS, Google, Salesforce)
- Windows Account
- z/OS Mainframe
- z/OS Mainframe (Priv Account)

* Does not require an Advanced Scripting Add-On License. Will require PowerShell installation.

** Professional Edition add-on/Platinum Edition only

Other platforms that Secret Server can change passwords on include:

- AS/400
- Linux / Mac
- Check Point
- Enterasys
- Dell DRAC
  - May require Pro Services, and is not covered by Delinea Support.

Privileged Account Credentials and Associated Secrets

By default, the Remote Password Changer (RPC) uses the credentials stored within a secret to initiate a password change. For Windows and Active Directory accounts, you can opt to use a privileged account by selecting the **Privileged Account Credentials** option. This allows you to choose an Active Directory secret that has the necessary permissions to change the account's password.

For secret templates that use a custom command password type, you can assign multiple **Associated Secrets** for use within the custom commands. When a secret is linked with Privileged account credentials or Associated Secrets, editing the username, host, domain, or machine is restricted for users who do *not* have access to those linked secrets. In the RPC tab, users without access will see the message "This Secret references another Secret for Remote Password Changing to which you do not have access. You will not be able to edit some fields on this Secret". Additionally, on the **Edit** page, all text-entry fields mapped for RPC, except for the password, are disabled. This added layer of security prevents unauthorized users from altering the username and resetting another account's password.

To appear in searches, privileged accounts must have RPC enabled in their originating template or the Active Directory secret.

If you attempt to use a secret that has checkout enabled as a Privileged Account on another secret, that second secret's password change will fail with an error that indicates that the associated secret has checkout enabled.

RPC Procedure

Various procedures related to RPC.

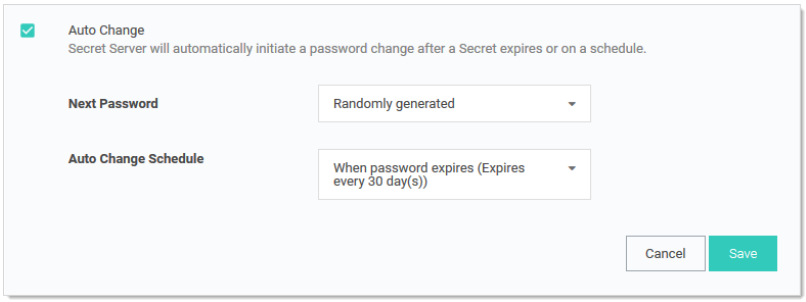
Automatic Remote Password Changing

The Remote Password Changing tab contains the settings for configuring RPC on an individual secret. Enabling RPC *auto change* on a secret allows Secret Server to remotely change the password when it expires. The user must have owner permission on the secret to enable auto change.

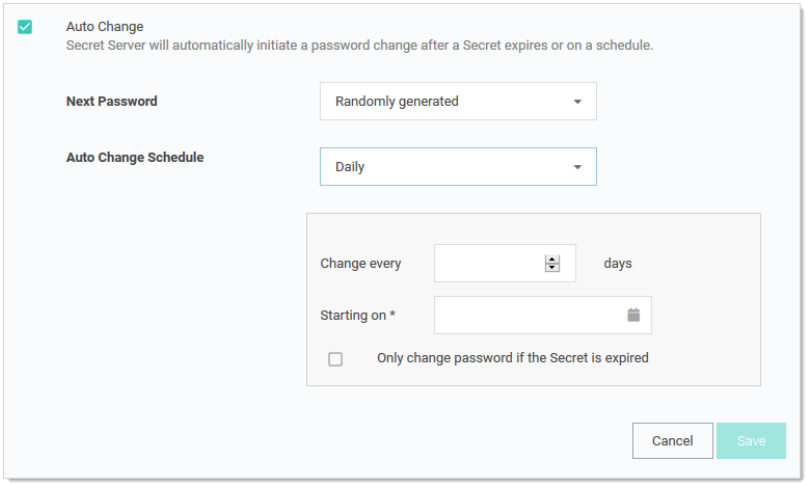
> If the password change fails, Secret Server flags the secret as out of sync and continue to retry until it is successful. If the secret cannot be corrected or brought In sync, manually disabling auto change stops the secret from being retried.

Auto Change Schedule

The Auto Change Schedule button is visible on the secret View RPC tab when RPC and autochange is enabled on a secret.



The Auto Change Schedule section, which appears when you set the Auto Change Schedule list box to other than "When password expires," allows you to specify an interval (daily, weekly, or monthly), start date, start time, and time frame (interval count) for when the password can be changed:



This setting is useful for having the RPC occur during off-hours in order to prevent disruptions. By default, this setting is "When password expires," which allows the secret to be changed immediately upon expiration.

There is a check box in the auto change schedule settings labeled "Only change password if the secret is expired." When it is enabled, auto change will not change the password until after the secret expires. The auto change occurs on the first scheduled time after the secret expires. If the box is unchecked, auto change occurs on the defined schedule, whether or not the secret has expired.

While the password change is waiting for this next scheduled time, the RPC Log (visible by navigating to **Configuration > Remote Password Changing**) reports the secret could not be changed because of a time schedule. The secret also remains expired until this auto change schedule is reached, even if the secret was forced to expire.

Understanding Expiration, Auto Change and Auto Change Schedules

Definition

What is the difference between expiration, auto change and auto change schedules?

- **Expiration**: sets whether or not a secret in Secret Server is marked as expired and the period Secret Server counts down before marking the secret as expired.
- **Auto Change**: sets Secret Server to automatically initiate a password change after a secret expires.
- **Auto Change Schedule**: sets the day and time to initiate the password change after the secret has expired. This cannot be configured without also enabling Auto Change.

Examples

Some examples to illustrate this:

Scenario One: Expiration with Auto Change and No Auto Change Schedule

- A Secret has an expiration period of 30 days, and auto change is enabled. No auto change Schedule has been set.
- At the end of the 30-day expiration period, the secret will expire.
- Immediately after the secret expires, it will be queued for a password change.
- Once the password has been changed, the secret is no longer marked as expired and expiration is reset to count down again from 30 days.

Scenario Two: Expiration with Weekly Auto Change

- A secret has an expiati period of 30 days, auto change is enabled, and an auto change schedule is configured for Weekly, recurring once a week on on Tuesday, changing at 0300.
- At the end of the 30-day expiration period, the secret will expire.

- Immediately after the secret expires, Secret Server will comply with the auto change schedule to determine when a password change occurs.
- The secret is queued for a password change as soon as it becomes 0300 on a Tuesday.
- Once the password is changed, the secret is no longer marked as expired. Expiration is reset to count down again from 30 days.

Scenario Three: Expiration with No Auto Change

- A Secret has an expiration period of 30 days, and auto change is not enabled.
- At the end of the 30-day expiration period, the secret expires.
- The secret remains expired until the field it applies to (usually the password field) is updated on the secret. This happens by manually updating the field or by using the "Change Password Remotely" button on the Remote Password Changing tab of the secret.
- Once the password is changed, the secret is no longer be marked expired, and expiration is reset to count down again from 30 days.

Important Considerations and Best Practices

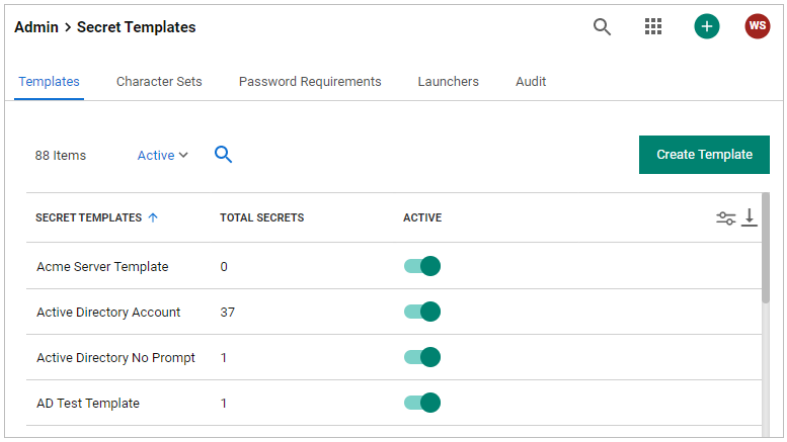
- If you want to rely strictly on expiration for password changing, enable auto change but set the schedule to none. Leave "Only change password when Secret is expired" checked.
- If you want to set an auto change schedule to run daily at a specific time, the change will only happen at maximum once per day at that given time. If a change happens already within that same day for the same secret, you cannot adjust the auto change schedule to run later within the same day and then have a password change occur again within that same 24-hour period. For example, if the password was already changed earlier in the day. The schedule is then adjusted to run a few minutes later within the same day. In that case, another password change will not occur until 24 hours has passed since the last change.
- If you set the auto-change schedule to run once per week, for example, on a Thursday, and "Only change password when secret is expired" is checked. Even if the secret expires on a Monday, a password change would not occur until the secret has expired and the scheduled time on Thursday has passed.
- If you set the auto change schedule to run once per week on a Thursday and "only change password when Secret is expired" is not checked, the password would be changed every Thursday, regardless of the secret's expiration status.
- If a secret has an expiration period but auto change is not enabled, no password change occurs automatically. The expiration would only update when the password is manually updated or a remote password change is manually triggered through Secret Server.
- If you want to change a password more frequently than once per day, we recommend using some of the advanced security features at the secret level or controlling the change through a secret policy. Use the check out feature combined with "Change Password on Check In" on the Security tab of a Secret. You can specify a custom interval to check out the secret. After the password check out interval expires or a user manually checks in the secret, the password is automatically changed.

For the configuration above, ensure that these accounts have a password-related group policy in Active Directory that specifies that the "Minimum Password Age" is set to 0. We recommend creating fine-grained password policies to achieve this. Add all the accounts that need rotation more frequently than once per day to an AD security group assigned to the fine-grained password policy. See Password and account lockout policies on Azure Active Directory Domain Services managed domains for more information.

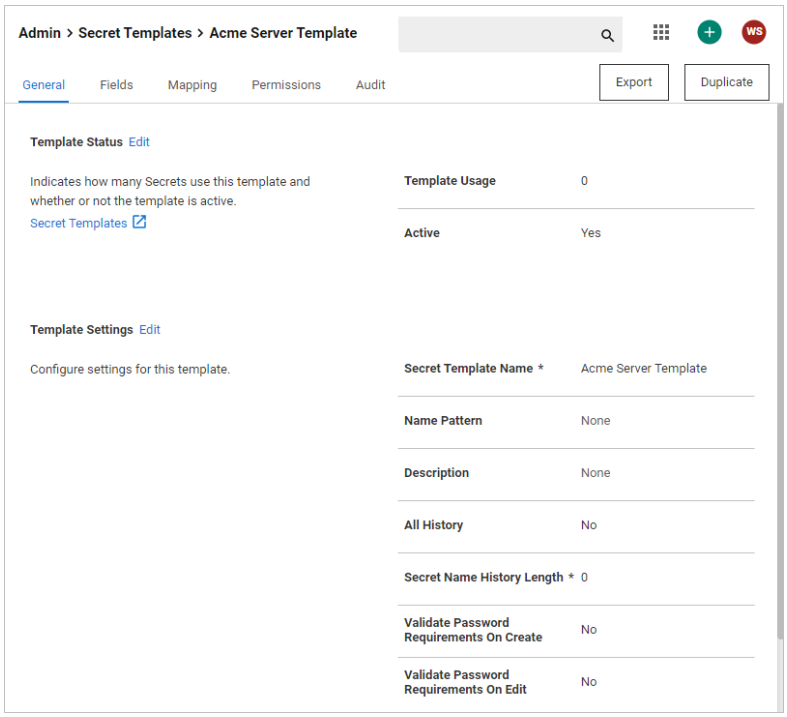
Assigning a Password Changer to a Secret Template

To assign any type of password changer to a Secret template, use the procedure below.

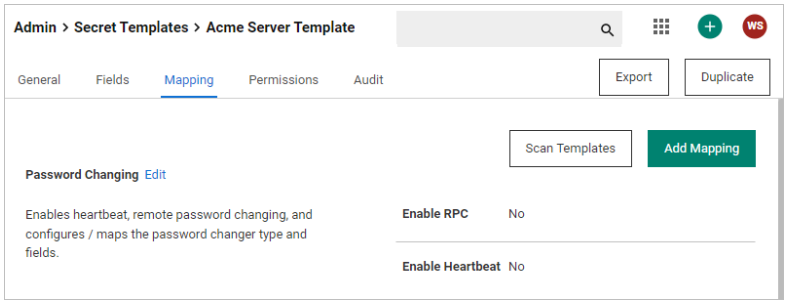
1. From the **Admin** menu, select **Secret Templates**. The Secret Templates page appears:



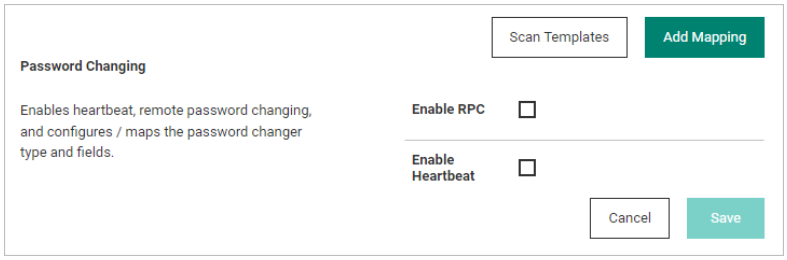
2. Click the name of the desired template in the list. The page for that template appears:



3. Click the **Mapping** tab:

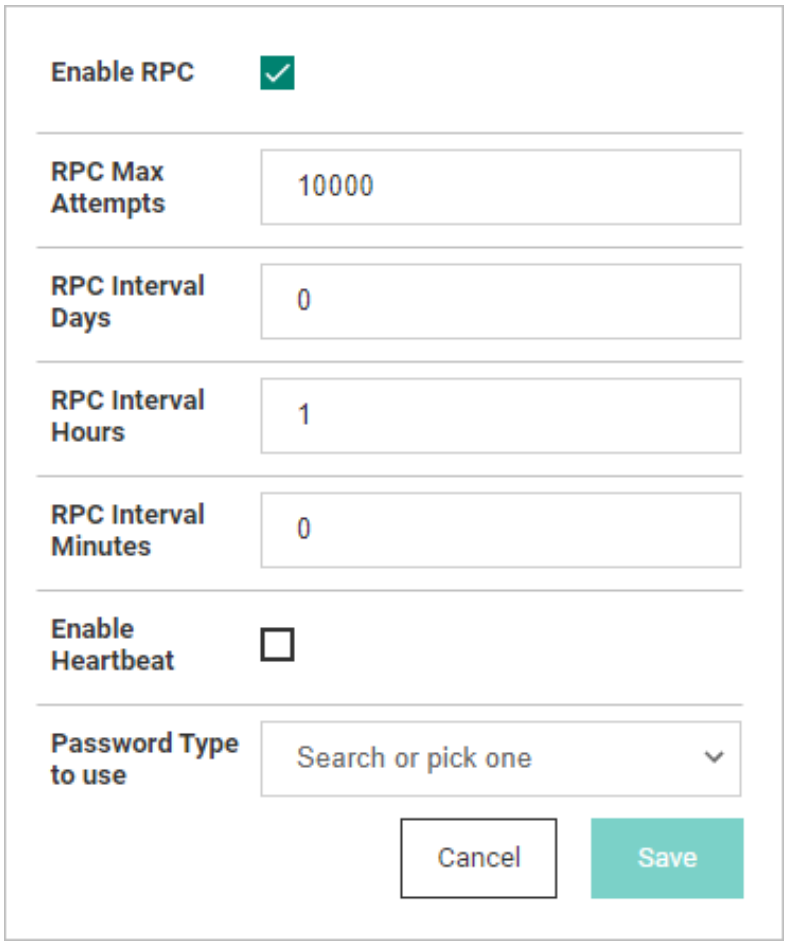


4. Click the **Edit** link next to the **Password Changing** section. The section becomes editable:

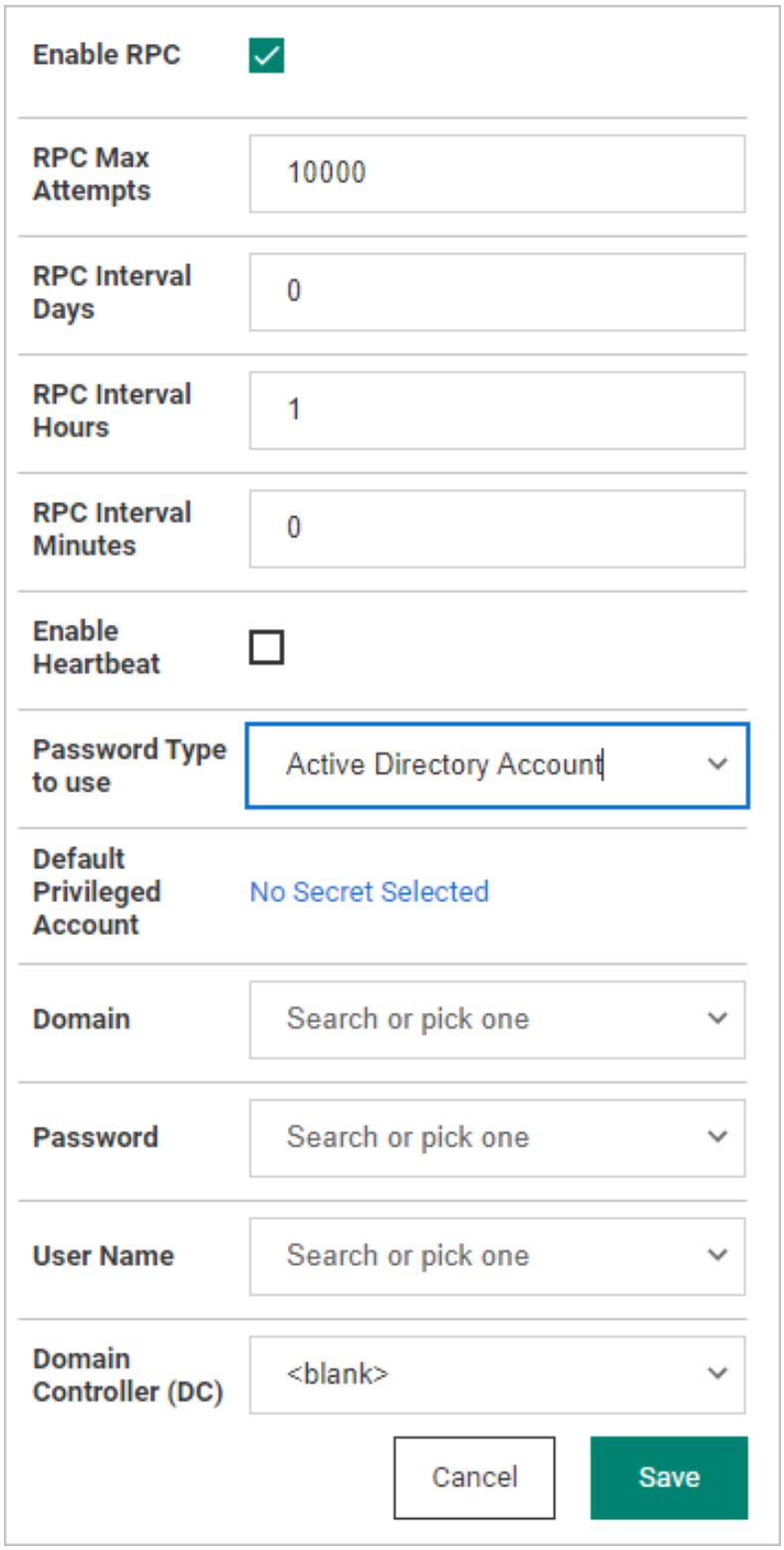


5. Click to select the **Enable RPC** check box. An editable section appears:

> This enables the feature, but auto RPC is set through policies or secret settings.



6. Click the **Password Type to Use** dropdown list to select the desired type. For this instruction, we chose Active Directory Account. New controls appear for that specific type:


Enable RPC
RPC Max Attempts
10000
RPC Interval Days
0
RPC Interval Hours
1
RPC Interval Minutes
0
Enable Heartbeat
Password Type to use
Active Directory Account
Default Privileged Account
No Secret Selected
Domain
Search or pick one
Password
Search or pick one
User Name
Search or pick one
Domain Controller (DC)
<blank>
Cancel
Save

7. Click to select or type text for each item to configure your desired setup. For more on configuring your choice, refer to the applicable configuration in the "RPC Overview" on page 904 section for details. For example, for Active Directory, go to "Configuring Delegation Control for the Administrative Account" on page 1000.
8. When finished, click the **Save** button.

Changing Ports and Line Endings

To change the port or line ending used on a password changer, click the password changer on the **Configure Password Changers** page and then click **Edit**. There, you can choose the line ending and port used by the device. By default, line endings are set to New Line (\n), however some devices and applications (such as HP iLO) use a different line ending system. The port defaults to 22 for SSH connections and 23 for Telnet connections.

For the built in Windows password changer there is a ports text-entry field available that can be filled in to help ensure a computer is listening. This can be used if DNS returns multiple IP addresses for a single box and only one is valid. For example, a laptop might get two IP addresses for an Ethernet and wireless connection, but if it is unplugged the Ethernet IP is invalid. In this case, Secret Server can do a reverse lookup and test each IP until it is able to connect on one of the specified ports. When it gets a response, it uses that IP for the password change.

Creating RPC Scripts

PowerShell scripts, SSH scripts, and SQL scripts for password changing (PowerShell only), dependencies, and discovery custom actions can be created by administrators with the role permission called Administer Scripts. The scripts can be accessed by going to **Administration > Remote Password Changing > Scripts**.

Secret Server requires that WinRM is configured on the Web server. For instructions please see "Configuring WinRM for PowerShell" on page 1483.

Creating Scripts

On the **Scripts** screen, select desired script tab and click **Create New** to enter the name of the script, a description, and the commands to run, then click **OK**. The script now shows up in the grid. Scripts can be deactivated and reactivated from the grid.

Testing Scripts

All scripts run from the machine that Secret Server is installed on, or the site assigned to the secret. To test a script, click the **Test** button on the grid next to the corresponding script.

PowerShell scripts run as the identity of the secret, so enter in an Active Directory credential to run the script as or select a secret to pre-fill the run-as credentials. Then enter in any command line arguments that the script requires. The output of the script is displayed above the grid for debugging purposes. To test the script over an engine, select a site from the **Site** list. This helps in diagnosing server specific issues where engines are installed.

Using Scripts

To use a script as a password changer or Dependency, it must be wired up to the appropriate action under **Admin > Remote Password Changing** on the password changer or dependency changer.

Discovery scripting is done under **Admin > Discovery > Extensible Discovery**. For more information on configuring extensible discovery see "Extensible Discovery" on page 571.

Viewing Audits

A full history of each PowerShell script is kept and can be downloaded from the audit trail. Click **View Audit** to view the audit trail for PowerShell. Each time a script is updated, the previous one can be downloaded from the corresponding audit record.

> For additional information on setting up PowerShell scripts, please read the following topic: "Creating and Using PowerShell Scripts" on page 1475.

Deactivating Password Changers

To make a password changer unavailable for use and to hide it from view in your list of password changers, you must mark it inactive:

1. From the **Password Changers Configuration** page, click the password type name of the password changer you would like to make inactive.
2. Click **Edit**.
3. Uncheck the **Active** box.
4. Click **Save**.

To view inactive password changers, check the **Show Inactive** box at the bottom of the list of password changers. The Active column in the table indicates the status of the password changer.

Enabling RPC

RPC is enabled under the Administration, Remote Password Changing page. Click **Edit** to enable RPC, secret heartbeat, and secret checkout. Once enabled, all secret templates with RPC configured are available to use with RPC.

Modifying Password Changers

To modify a password changer, click the password changer name under **Admin > Remote Password Changing > Configure Password Changers** and then use the **Edit** or **Edit Commands** buttons to make changes. For more information about editing the custom PowerShell password changer, see "Running RPC with PowerShell" on the next page.

> You can find the full, up-to-date list of password changers included with Secret Server by default in "Password Changer List" on page 909.

Running a Manual RPC

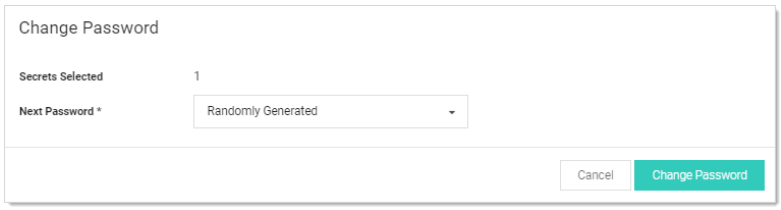
On the RPC tab there is a button called Change Password Remotely button that allows the use to change the password immediately instead of waiting for it to expire. When this button is clicked the user is taken to the Change Password Remotely page where they can enter in or generate the new password for the account. When the user clicks the Change button the secret enters the queue for having its password changed. The RPC Log found on the Remote Password Changing page details the results of the password change attempts and can be used for debugging.

If the secret is a Unix or Linux account and uses a password changer that supports SSH key rotation, the user can change the account's password, public and private keypair, and the private key passphrase. The user can enter or generate any of these items.

> If the password change fails, Secret Server continues to retry until it is successful, or the change is canceled by the user. To manually cancel the change, click Cancel Password Change on the RPC tab.

To run a manual RPC:

1. From **Dashboard**, click its check box to select secret you want to test.
2. Click the Change Password Remotely button. The Change Password popup page appears:



3. Click to select the **Next Password** dropdown list and select **Manual** or **Randomly Generated**. If you chose manual:
   a. The Password text box appears.
   b. Type the new password in the **Password** text box.
   c. Click the **Change Password** button.

   Otherwise, click the **Change Password** button. The password change is now queued.
4. You can verify that the password change completed either by unmasking the password on this screen (click the lock icon beside the password field) or by looking at the **Remote Password Changing** log. You can find the Remote Password Changing log by going to **Admin > Remote Password Changing**.

Running RPC with PowerShell

Overview

Secret Server supports running PowerShell scripts for Remote Password Changing (RPC) and heartbeat. Below are the steps for creating an Active Directory (AD) password changer that uses PowerShell scripts. The example is meant as a simple guide for how to wire-up the template to scripts as a proof of concept. Your actual PowerShell password changer scripts may be more complex depending on your environment and needs.

> Before you begin, please ensure password changing and heartbeat are enabled in **Admin > Remote Password Changing** and review the information on "Configuring CredSSP for WinRM with PowerShell" on page 1479, which will be necessary for most PowerShell password changing tasks.

> Do not use quotation marks (") in passwords for PowerShell scripts for RPC. Quotation characters are dropped from passwords when passing the $PASSWORD token to a custom password changer PowerShell script, resulting in an authentication failure.

Procedure

The PowerShell scripts are created and accessed through the **Admin > Scripts** page. To create a PowerShell password changer, you need to create two scripts. The first script verifies the account's current password. The second script changes the account's password. These two scripts are linked to a new secret template.

Task 1: Creating the Active Directory Verify Password Script

1. Navigate to **Admin > Scripts**.
2. Click the **+Create New** button on the **PowerShell** tab.
3. Type the following information in the dialog:
   - **Name**: Active Directory Verify
   - **Description**: Script used to verify an Active Directory account
   - **Category**: Heartbeat
   - **Script**:

```
$domain = "LDAP://"+$Args[0];
$dn = New-Object System.DirectoryServices.DirectoryEntry($domain, $Args[1], $Args[2]);
if ($dn.name -eq $null){ throw "Authentication failed - please verify your username and
password." };
```

4. Click the **OK** button to save the script.

Task 2: Creating the Active Directory Change Script

1. On the **PowerShell** tab, click the **+ Create New** button.
2. Type the following information in the dialog:
   - **Name**: Active Directory Change
   - **Description**: Script used to change the password of an Active Directory account
   - **Category**: Password Changing
   - **Script**:

```
$Domain = $args[0]
$UserToChange = $args[1]
$NewPassword = $args[2]
$P_User = $args[0] + "\" + $args[3]
$P_PWord = ConvertTo-SecureString -String $args[4] -AsPlainText -Force
```

```
$Creds = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $P_
User, $P_PWord
$pwd = ConvertTo-SecureString $NewPassword -AsPlainText -Force;
Set-ADAccountPassword -Server $Domain -Identity $UserToChange -NewPassword $pwd -Reset -
Credential $creds
```

3. Click the **OK** button to save the script.

Task 3: Testing the Scripts

For the AD verification script:

1. Go to **Scripts > PowerShell tab**.
2. Click the Run Script arrow icon on the AD verify script. The Test Script popup appears.
3. Type the arguments (separated by spaces) in the **Arguments** text box: domain name (for you), username (yours), password (yours). For example: `my.company.com ssadmin FD#@789Uik4$`
4. Type your domain name for the script-running account in the **Domain** text box.
5. Type the username in the **Username** text box for account that can run PowerShell scripts on the domain.
6. Type that user's password in the **Password** text box.
7. Click the **OK** button to test your script the with provided parameters.

For the Active Directory change script:

1. Go to **Scripts > PowerShell tab**.
2. Click the Run Script arrow icon on the AD change script. The Test Script popup appears.
3. Type the arguments (separated by spaces) in the **Arguments** text box: domain name (for you), username (yours), new password (yours), domain admin username, domain admin password. For example: `my.company.com ssuser O8sSKthsoidPW ssadmin FD#@789Uik4$`
4. Type your domain name for the script-running account in the **Domain** text box.
5. Type the username in the **Username** text box for account that can run PowerShell scripts on the domain.
6. Type that user's password in the **Password** text box.
7. Click the **OK** button to test your script the with provided parameters.

If successful, this will change the password on the account that is used for testing.

The remaining steps depend on the version of Secret Server you are using. In Secret Server 10.0.000006 we introduced the ability to create multiple PowerShell password changers, each with their own set of password change and verify scripts. These password changers can be assigned to different scan templates to automatically assign different PowerShell password changer scripts to different types of local accounts when creating local account import rules in discovery. For more information about how scan templates and password changers are used in discovery and local account import rules, please see "Discovery Overview" on page 522. Prior to 10.0.000006, there was only one PowerShell password changer and the scripts were assigned on the secret template.

Task 4: Configuring a Password Changer for Secret Server Version 10.0.000006 and Later

In Secret Server versions 10.0.000006 and later, after the scripts are tested and working correctly, the next step is to create a PowerShell password changer.

1. Go to **Admin > Remote Password Changing**.
2. Click the **Configure Password Changers** button.
3. Click the **New** button.
4. In the **Base Password Changer** dropdown list, select **PowerShell Script**.
5. Type the name of the new password changer.
6. Click the **Save** button. On the next page you will select the scripts to use for password changing and verification (heartbeat).
7. Under **Password Change Commands**:
   a. Select the script that you created to do password changes.
   b. Type the following in the **Script Args** text box: `$DOMAIN $USERNAME $NEWPASSWORD $[1]$USERNAME $[1]$PASSWORD`.
   c. Click the **Save** button next to the **Script Args** text box.
8. Under **Verify Password Changed Commands**:
   a. Select the script that you created to do heartbeats and verification.
   b. Type the following in the **Script Args** field: `$DOMAIN $USERNAME $PASSWORD`.
   c. Click the **Save** button next to the **Script Args** text box.

When Secret Server runs the script, it replaces the fields with the matching secret field values. $NEWPASSWORD is a special case for the new password that is generated by Secret Server or specified by the user when performing a password change. For more information see "Using Secret Fields in Scripts" on page 1494.

You must specify scripts for both sections and you must click the Save button next to each one for both to save.

Task 5: Creating a Secret Template

The next step is to create the secret template:

1. Go to **Admin > Secret Templates**.
2. Click the **Create New** button.
3. Name the template `PowerShell Active Directory`.
4. Create the following new fields:

- Domain Field Type: Text
- Username Field Type: Text
- Password Field Type: Password
- Notes Field Type: Notes

5. Click the **Configure Password Changing** button.
6. Click the **Edit** button.
7. Click to select the **Enable Remote Password Changing** and **Enable Heartbeat** checkboxes.

Task 6a: Finishing the Secret Template Configuration for Secret Server 10.0.000006 and later

Complete either 6a or 6b, not both.

1. Select the password changer created in the previous section from the **Password Type to use** dropdown list.
2. Click to select **Domain** next to the **Domain** field.
3. Click to select **Username** next to the **User Name** field.
4. Click to select **Password** next to the **Password** field.
5. Click the **Save** button to save the mapping.

Task 6b: Finishing the Secret Template Configuration for Secret Server 8.8.000000 to 10.0.000000

Complete either 6a or 6b, not both.

1. Select **PowerShell Script** from the **Password Type to use** dropdown.
2. Click to select **Domain** next to the Domain field.
3. Click to select **Username** next to the User Name field.
4. Click to select **Password** next to the Password field.
5. Click to select **Active Directory Change** next to the **Remote Password Change Script** field.
6. Enter the following to the **Remote Password Change Args** field: `$DOMAIN $USERNAME $NEWPASSWORD $[1]$USERNAME $[1]$PASSWORD`.
7. Click to select **Active Directory Verify** next to the **Heartbeat Script** field.
8. Type the following next to the **Heartbeat Args** field: `$DOMAIN $USERNAME $PASSWORD`.

   When Secret Server runs the script, it replaces the fields with the matching secret field values. $NEWPASSWORD is a special case for the new password that is generated by Secret Server or specified by the user when performing a password change.

9. Click the **Save** button to save the mapping.

Task 7: Creating Secrets Using PowerShell Remote Password Changing

Create the AD account secret PowerShell account:

1. Create three secrets (The first two **must** be different secrets):
   - One that is an Active Directory Account that has the necessary rights to run PowerShell on your domain
   - One that is an Active Directory Account that has the necessary rights to run a password change on your domain
   - One that is based on the new PowerShell Active Directory Template.
2. Create the Active Directory account secret PowerShell account.
3. On the dashboard, use the dropdown on the **Create Secret** widget and select **Active Directory Account**. Use the following parameters:
   - **Secret Name:** PowerShell Admin
   - **Domain:** Domain that the account exists on
   - **Username:** Account name that can run PowerShell scripts in the domain
   - **Password:** Password for the account
4. Click the **Save** button to save your secret and verify that it passes heartbeat.
5. Click the **Home** button to return to the dashboard.

Create the AD account secret for password changing:

1. On the dashboard, use the dropdown on the **Create Secret** widget and select **Active Directory Account**. Use the following parameters:
   - **Secret Name:** Password changing Admin
   - **Domain:** Domain that the account exists on
   - **Username:** Account name that can change passwords in the domain
   - **Password:** Password for the account
2. Click the **Save** button to save your secret and verify that it passes heartbeat.
3. Click the **Home** button to return to the dashboard.

Create the PowerShell Active Directory secret:

1. On the dashboard, use the dropdown on the **Create Secret** widget and select **PowerShell Active Directory Account**. Use the following parameters:
   - **Secret Name:** PowerShell AD user
   - **Domain:** Domain that the account exists on
   - **Username:** samAccountName of the account to be managed
   - **Password:** Password for the account
2. Click the **Save** button to save your secret and verify that it passes heartbeat.

3. Click the **Remote Password Changing** tab for the secret.
4. Click the Edit button.
5. Click to select **Privileged Account Credentials** in **Execute PowerShell**. The Privileged Account selector appears.
6. Click the **No Selected** Secret link.
7. Locate click on the **PowerShell Admin** secret.
8. Click the **Home** button to return to the dashboard.
9. In the **The following Secrets are available to be used in Custom Password Changing Commands and Scripts** section:
    a. Click the **No Selected Secret** link.
    b. Select your AD account secret for password changing.
    c. Click on the **Save** button.

Everything should now be configured for heartbeat and RPC on the Secret. Run **Heartbeat** (from the **General** tab in the Secret) to confirm that it works and run an RPC ** (from the **Remote Password Changing** tab of the secret) to confirm that it also works.

Errors

If you receive the "The term 'Set-ADAccountPassword' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again." error, install the AD-Domain-Services in PowerShell. To do this start PowerShell as an administrator then run the following command:

```
Install-windowsfeature -name AD-Domain-Services –IncludeManagementTools
```

Additionally you may need to install the Remote Server Administration Tools for your version of Windows and then in PowerShell run:

```
Import-Module Servermanager
```

Using RPC for Secrets with Shared Credentials

In most environments, we recommend using a separate password for each account for optimal security. However in environments where identical credentials are used in multiple secrets, we recommend using RPC to change the password on one primary parent account secret, and then using a PowerShell dependency script to update values in child secrets. The PowerShell script calls back to Secret Server's API, retrieves a list of comma-separated values representing child secret IDs, and updates the values stored in the child secrets. We recommend using this process for no more than 25 child secrets.

Requirements

- A Secret Server instance version 10.1.000000 or newer with a premium add-on or Enterprise Plus
- A PowerShell implementation enabled and working properly. See "Configuring WinRM for PowerShell" on page 1483
- The PowerShell Wellness Checker

For this procedure you will need to create the four types of user accounts listed below, and for each account you will need to create a corresponding secret in Secret Server with the account's login credentials and other information.

Create the user accounts and secrets described below:

- An API User account and a corresponding secret. This API User account will NOT take up a user license. Recommended templates for the secret include the Active Directory template and the Web Password template. Credentials may be a local account or an Active Directory service account assigned to the Synchronization group, but must be stored in Secret Server to be passed to the PowerShell script.
- A primary parent account and a corresponding secret that has RPC set up and the PowerShell dependency script from this page attached. The primary parent account credentials may be either a local account or an Active Directory service account assigned to the Synchronization group.
- Child accounts with a corresponding secret for each account containing the child secret ID, with edit permissions granted to the API User account.
- A privileged Active Directory account and a corresponding secret that can run PowerShell on the Secret Server machine.

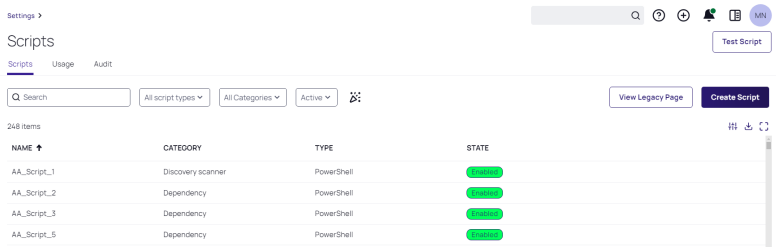
To create a new dependency changer for synchronizing passwords during RPC, first:

1. On the server that the script will be processed from, whether DEs or web nodes, download the WellnessChecker tool ZIP file.
2. Extract the ZIP file and run this command:

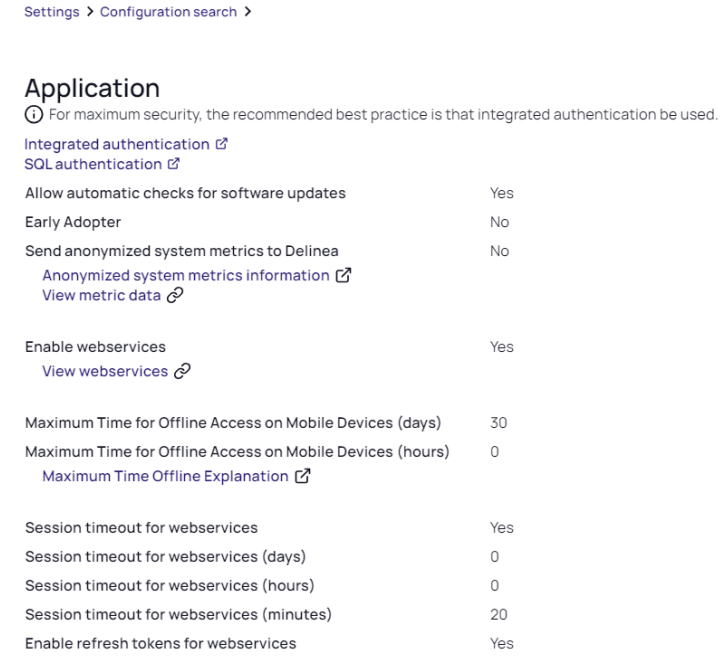
```
PowerShell.WellnessChecker.exe -fixerrors
```

Then follow the steps below:

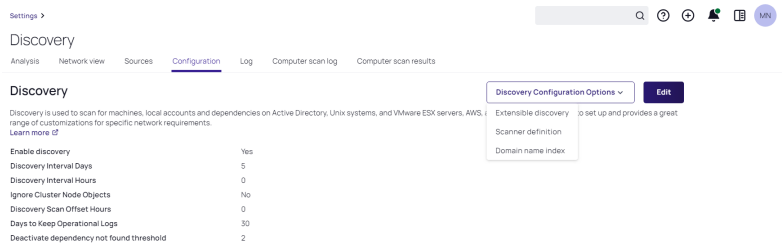
1. In Secret Server, browse to **Admin > Scripts**.
2. Click **Create Script**.



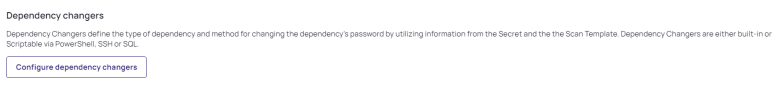
3. In the **New Script** dialog, fill in the fields for **Name** and **Description**, for the **Script Type** select **PowerShell** and specify**Category**.
4. Check **Enabled** next to the **State** option.
5. In the **Script** field, paste in the script provided at the bottom of this page.
6. Click **Save** at the bottom of the page to save the file.
7. Browse to **Admin > Application**, and make sure **Enable Webservices** is set to **Yes**.



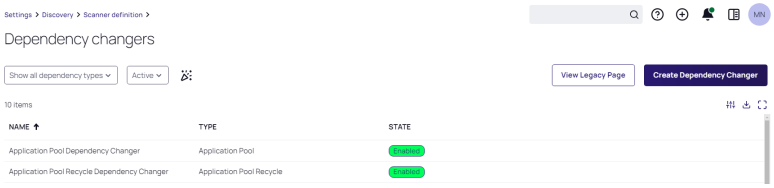
8. Browse to the primary parent account secret and ensure that RPC is setup on it.
9. In the primary parent account secret, click the **RPC** tab.
10. Click **Edit**.
11. In the secret grid at the bottom, select the API User account secret you created. The API User account secret should be the only secret in the grid.
12. Browse to **Admin > Discovery** and click the **Configuration** tab.
13. Click **Discovery Configuration Options** and select **Extensible Discovery** from the drop-down list.



14. On the **Extensible Discovery Configuration** page, click **Configure Dependency Changers**.

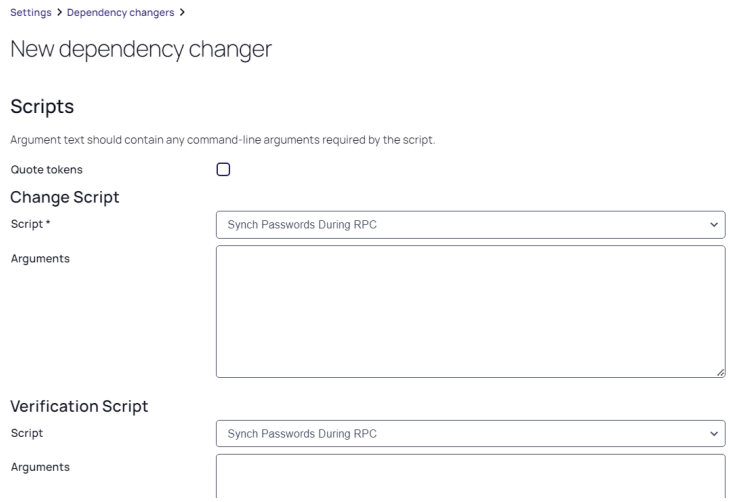


15. On the **Secret Dependency Changers** page, click **Create Dependency Changer**.



16. In the **New Dependency Changer** dialog, enter the **Name**, check the **State** to **Enabled**, for **Dependency Type** select **PowerShell script**, for the **Scan template** select **Computer Dependency (Basic)**, check the **Create template** checkbox.

17. In the **Scripts** section below enter the related **Change Script**, **Verification Script**, **Change Success Script**, and **Change Fail Script**.



18. In the **Arguments** field, paste the following:

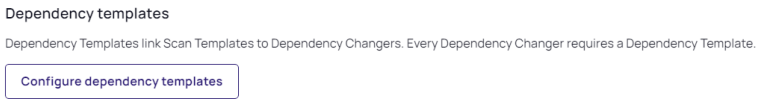
    `$[1]$USERNAME $[1]$PASSWORD $PASSWORD $NOTES $[1]$DOMAIN`

    The actions of the Arguments are as follows:

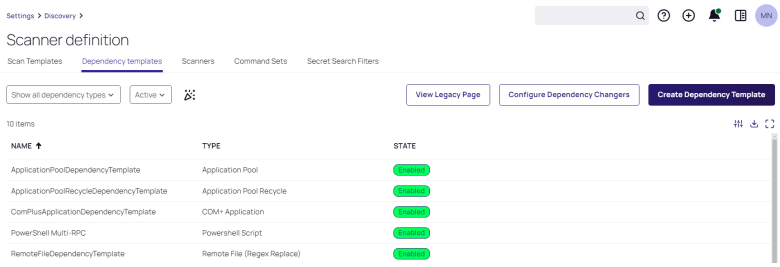
    - `$[1]$USERNAME` pulls the username from the privileged account on the primary parent account, which will be used to execute the PowerShell script.
    - `$[1]$PASSWORD` pulls the password from the associated secret on the primary parent account, which will be used to execute the PowerShell script.

- `$PASSWORD` pulls the password from the primary parent account, which will be set for all secrets listed in the **Notes** field.
- `$NOTES` pulls the **Notes** content from the primary parent account, and parses the comma separated list of secret IDs to find the other secrets to update.
- `$[1]$DOMAIN` pulls the **Domain** field from the associated secret on the primary parent account. For local accounts, leave the **Domain** field on the associated secret empty. It must be listed last because of the way PowerShell parses empty fields.

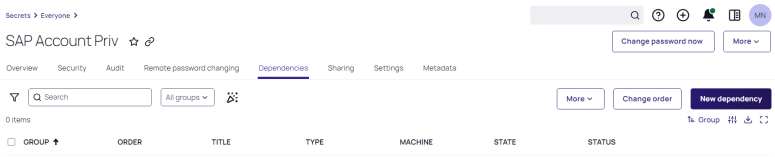
19. Browse back to the **Extensible Discovery Configuration** page and this time, click **Configure Dependency Templates**.



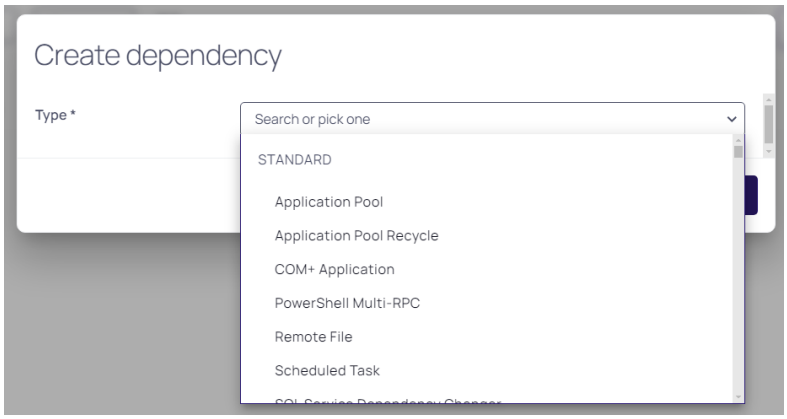
20. On the **Secret Dependency Templates Designer** page, select the new dependency changer you configured in the last step.



21. Browse to the primary parent account secret and click the **Dependencies** tab.
22. Click **New Dependency**.



23. In the **Create dependency** dialog, click the **Type** dropdown and select the PowerShell dependency template you created.



24. In the **Create dependency** dialog, select **Dependency group**, enter **New group name**, enter **New group site name**, enter **ServiceName**, check the **Enabled** checkbox, enter `default` in the **Machine Name** field, and click **Save**.
25. In the primary parent account secret's **Notes** field, ensure that the child secret IDs appear in a comma-separated-values list, for example `19,39,81...`

Now the dependency has been added and you can test the full process by running a remote password change on the primary parent account. All of the secrets listed by ID in the **Notes** field should be updated with the same password.

PowerShell Script

> Do not use quotation marks (") in passwords for PowerShell scripts for RPC. Quotation characters are dropped from passwords when passing the $PASSWORD token to a custom password changer PowerShell script, resulting in an authentication failure.

Replace `$url` with the name of the machine hosting your Secret Server instance.

```
$url = 'https://MySecretServerURL/webservices/sswebservice.asmx';
$username = $Args[0]
$password = $Args[1]
$newpassword = $Args[2]
$secretIdArray = $Args[3]
$domain = $Args[4]
$proxy = New-WebServiceProxy -uri $url -UseDefaultCredential
$result1 = $proxy.Authenticate($username, $password, '', $domain)
if ($result1.Errors.length -gt 0){
    $errors = $result1.Errors[0]
    Write-Debug "Errors result1: $errors"
    exit
} else {
    $token = $result1.Token
}
$secretIds = $secretIdArray -split ","
foreach($secretId in $secretIds){
```

```
    $result2 = $proxy.GetSecret($token, $secretId, $false, $null)
    if ($result2.Errors.length -gt 0){
        $errors = $result2.Errors[0]
        Write-Debug "Errors result2: $errors"
    } else {
        $secretName = $result2.Secret.Name
        Write-Debug "Updating Secret: $secretName"
        foreach ($item in $result2.Secret.Items) {
            if($item.IsPassword) {
                $item.Value = $newpassword
            }
        }
        $secret = $result2.Secret
        $result3 = $proxy.UpdateSecret($token, $secret)
        if ($result3.Errors.length -gt 0) {
        $errors = $result3.Errors[0]
            Write-Debug "Errors result3: $errors"
        } else {
            Write-Debug "Updated Secret: $secretName"
        }
    }
}
```

Secret Dependencies for RPC

Secret dependencies are items that rely on the username, password, or SSH private key stored in the secret. By adding them to the Dependencies tab, they are automatically updated when the secret's password is changed, ensuring they are up to date with the account on which they depend.

Adding a custom dependency template may require additional settings (these settings are described in the subtopics in this section).

COM+ Dependency Scanner

The COM+ Dependency Scanner allows for an Active Directory domain discovery source to locate COM+ Applications running on machines on the domain that are being run by Domain Accounts.

Firewall concerns may be addressed by ensuring that Port 135 is open between the target machine being scanned and the machine that engine is installed on.

Requirements for Discovery

Windows Services

For all supported versions of Windows and Windows Server, ensure that **Remote Procedure Call (RPC)** and **Remote Procedure Call (RPC) Locator** services are running. To help prevent any errors that would stop the services, set the **Startup Type** to **Automatic**.

Component Services

For all supported versions of Windows and Windows Server, ensure that **NETWORK** has remote access permissions to the machine.

1. Open **Component Services** (dcomcnfg.exe).
2. Under **Console Root**, expand **Component Services** and the **Computers** folder.
3. Right-click **My Computer**
4. Select **Properties**.
5. Under the **Default Properties** tab, ensure that the **Default Authentication Level** is set to **Connect** and that **Default Impersonation Level** is set to **Identify**.
6. On the **COM Security** tab, for both the **Access Permissions** and **Launch and Activation Permissions** sections, click **Edit Limits** and then add **NETWORK**.
7. Check **Allow** for all Remote permissions.

If the **Edit Limits** button is disabled, open the **Local Security Policy**. Under **Security Settings** expand **Local Policies** and select **Security Options**. There will be two **DCOM: Machine Access/Launch Restrictions**. Edit the one that corresponds to the disabled **Edit Limits** buttons, adding **NETWORK** and giving Remote permissions there.

Editing or altering the existing permissions on the machine or editing the **Default Permissions** listed can have a negative impact on the machine.

COM+ Network Access

For all supported versions of Windows Server, ensure that COM+ Network Access is enabled by installing the Application Server Role. During the installation process, check the box next to **COM+ Network Access** under **Features**.

Versions Supported

- Windows 7
- Windows 8
- Windows 10
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2012 R2

Versions Not Supported

- Windows Vista and earlier versions of Windows
- Versions of Windows Server pre-2008
- Windows Server 2016

Configuring COM+ Discovery for a New Domain

1. Navigate to **Admin > Discovery**
2. Click **Edit Discovery Sources**.
3. Click **Create New**.
4. Select **Active Directory Discovery Source** and click **OK**.
5. In the wizard, click **Next**.
6. Select a **Site** that is set up with **Distributed Engine**
7. Click **Next**.

> The COM+ Dependency Scanner will only run when a Distributed Engine Site is applied to the Discovery Source. The Engine will need to be installed either on the Domain to be scanned, a Child Domain relative to the Domain being scanned, a Parent Domain relative to the Domain being scanned, or another Trusted Domain relative to the Domain being scanned.

8. Check the box next to **COM+ Application**.
9. Click **Next**.
10. Enter your **Fully Qualified Domain Name**.
11. Select or create a Secret for an Active Directory account that will scan for your COM+ dependencies.
12. Click **Next**.

Your new domain is now configured in Secret Server and Discovery will search for COM+ dependencies in it.

Configuring COM+ Discovery for an Existing Domain

1. Navigate to **Admin > Discovery**.
2. Click **Edit Discovery Sources**.
3. Click on the domain where you wish to search for COM+ dependencies.
4. Click the **Scanner Settings** tab.
5. Scroll down to the **Find Dependencies** section and click **Add New Dependency Scanner**.
6. Click the plus symbol to the left of **COM+ Application**.

   You will be unable to make additional changes.
7. Click **OK** to proceed. Discovery will now search for COM+ dependencies.

Creating Custom Dependencies

If there are different dependency types that you want to manage that are not supported out of the box, new ones can be created based on a script. A custom dependency consists of two components:

- **Dependency Template:** The dependency template defines how a dependency is matched to discovered accounts and how it updates the target after a password change occurs on the account. to create a new dependency template, go to **Admin > Secret Templates** and click the **Dependency Templates** button.
- **Dependency Changer**: A dependency changer is a script and the associated parameters to be passed into the script. Dependency changers can be created and modified by going to **Admin > Remote Password Changing > Configure Dependency Changers**.

> Please see "Discovery Overview" on page 522 for comprehensive guidance on configuring and using dependency changers and dependency templates.

Dependency Groups

By default, all dependencies are updated in the order listed. There are cases where you may want to split out different sets of dependencies into separate groups. Typically, this is because a single service account may run services across different segregated networks that can communicate with the domain but not each other and have different distributed engine sites assigned. In this case you can create two dependency groups and assign them to different distributed engine sites to solve connectivity issues.

Dependency Settings and Information

Dependencies have the following settings:

> Not all dependency types have all these settings.

- **Change Fail Script:** For SSH key rotation and SSH key rotation privileged dependency types, this is the built-in script that determines if Secret Server was unable to update the public key on the dependency.
- **Change Script:** For SSH key rotation and SSH key rotation privileged dependency types, this is the built-in script that updates the public key on the dependency.
- **Change Success Script:** For SSH key rotation and SSH key rotation privileged dependency types, this is the built-in script that determines if Secret Server was able to update the public key on the dependency.
- **Database:** For SQL script dependency types, the database name for the script.
- **Dependency Group:** Name of the group to run the dependency update in.
- **Description:** Description of the dependency for documentation purposes.
- **Enabled:** Whether Secret Server attempts to update the dependency. A disabled dependency is ignored by Secret Server.
- **File Path:** For Remote File Dependency types, this is the UNC file path on the remote server where the embedded password exists.
- **Machine Name:** Computer name or IP address on which the dependency is located.

- **Name:** Name of the dependency on the remote machine.
- **Port:** For SQL and SSH script dependency types, the port name for the script.
- **Privileged Account:** The account Secret Server authenticates as when changing the dependency's credentials, so it must have privileges on the remote machine to edit the dependency.
- **Public Key:** For SSH key rotation and SSH key rotation privileged dependency types, this text-entry field holds the value of the public key stored on the dependency.
- **Regex:** For Remote File Dependency types, the regular expression used to locate the password embedded in the configuration file.
- **Restart:** Determines if the dependency is restarted once the account has been updated.
- **Run Condition:** Allows the dependency to run conditionally depending on the outcome of the dependencies above it. Run Conditions are not applied when dependencies are run directly from the Dependency grid.
- **Script:** Name of the PowerShell script, SSH script, or SQL script in the scripts repository configured on the Dependency Template. The actual script selected can be previewed by clicking the eye icon.
- **Server Key Digest:** For SSH key rotation and SSH key rotation privileged dependency types, a text-entry field that serves as a security control for specifying the SHA1 hash of the SSH host key on the remote server.
- **Server Name:** For SQL script dependency types, the server name for the script.
- **SSH Key Secret:** An account with SSH Key that Secret Server uses to authenticate when executing the SSH Script or SSH Key rotation dependency types.
- **Template:** Whether the dependency is an IIS application pool, Scheduled Task, windows service, remote file, COM+ application. Custom dependencies can also be created using a SQL, SSH, or PowerShell script.
- **Verification Script:** For SSH key rotation and SSH key rotation privileged dependency types, this is the built-in script that verifies that the new public key on the dependency matches the private key on the secret.
- **Wait(s):** Time in seconds that Secret Server pauses before changing the dependency.

Example values for a Windows service dependency on a remote computer might be: `192.11.158.99`, `Windows Service`, `aspnet_state`, or `DOMAIN\admin`.

The following operations can be performed in the Dependency grid:

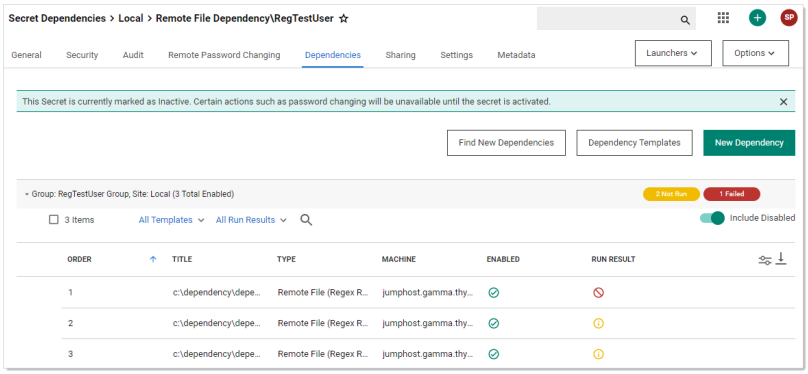
- **Edit:** Click on three dots next to a dependency name, and select Edit from the drop down. The Edit dependency window will open. In the Edit dependency window fill in:
  - **Type:** Select the related dependency type from the dropdown.
  - **Dependency group:** Select existing or add new dependency group.
  - **Sort order:** Select the related sort order.
  - **Description** Add the description if prompted.
  - **Enabled:** Check to enable dependency.
  - **Run As:** Click **No secret** selected to add a secret, or click on the selected secret to replace it. In the Select Secret window select the related secret to run the script and set the password on the dependency to the current password for the secret.

  - **Wait (s):** Set the number of seconds.
  - **Restart on Password Change:** Check to enable restart when password is changed.
  - **Machine name:** Enter the related machine name.
- **View log:** Click on three dots next to a dependency name, and select View log from the drop down to see the Secret dependency history with dates of events and related messages.
- **Enable:** Click on three dots next to a dependency name, and select Enable to enable the dependency, click OK to confirm. This will enable the selected Dependency. Alternatively, select a dependency (or several) from the list by checking them, and click Enable in the pop up above - in this way you can enable several dependencies at a time.
- **Disable:** Click on three dots next to a dependency name, and select Disable to disable the dependency, click OK to confirm. This will disable the selected Dependency. Alternatively, select a dependency (or several) from the list by checking them, and click Disable in the pop up above - in this way you can disable several dependencies at a time.
- **Delete:** Click on three dots next to a dependency name, and select Delete to delete the dependency, click OK to confirm. This will delete the selected Dependency. Alternatively, select a dependency (or several) from the list by checking them, and click Delete in the pop up above - in this way you can delete several dependencies at a time.
- **Run:** Select a dependency (or several) from the list by checking them, and click Run in the pop up above - this will run the selected Dependencies. Alternatively, click the return arrow icon next to a dependency to run it, the tests results are displayed afterward.

> Due to security constraints, scheduled tasks require an Active Directory domain user as the privileged account.

Dependency Status

You can see a list and status of all dependencies for a secret when viewing that secret in the UI. For example:



Account Dependence

You can use a single account (username and credential) for OS login and also for running Windows services, scheduled tasks, IIS application pools, and more. This is especially common for functional accounts. You can link

the credentials of one account object to such usages. The link is independent of the actual system that the dependency is running on.

For example, given this scenario:

- The Windows domain account DOM\svc_app1 is managed on windows domain controller, where it is located.
- SRV1 is running a Windows service under the user DOM\svc_app1.
- SRV2 is running a scheduled task under the user DOM\svc_app1.
- When the account DOM\svc_app1 is changed, the dependencies of that account on SRV1 and SRV2 need to be updated too.
- The Windows service on SRV1 may have to be restarted for the password change to work.

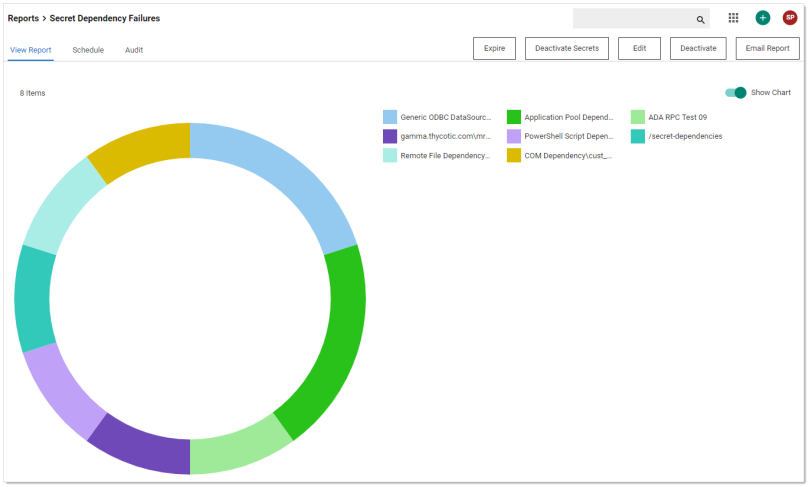
The modeling of the dependency would take place in / with the master object, which in this case is the windows domain account DOM\svc_app1. When selecting the master account object in the UI, it shows the dependencies along with their status.

Viewing Dependency Status

We offer four reports for viewing your secret dependency status:

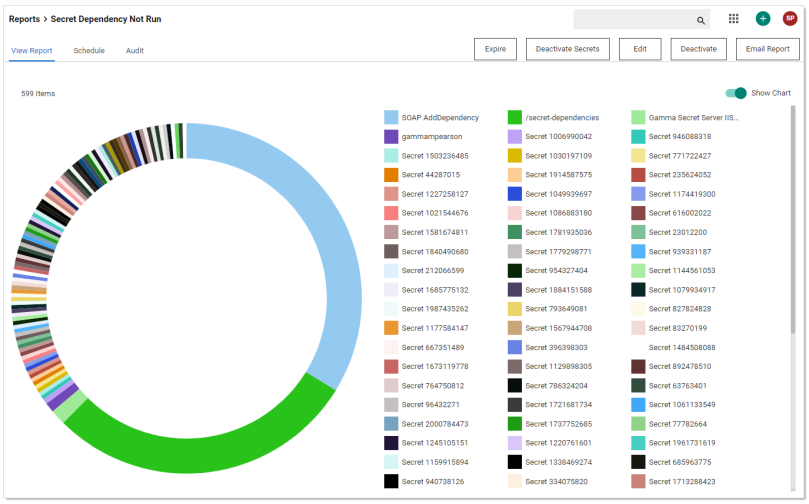
Secret Dependency Failures

Figure: Secret Dependency Failures Report



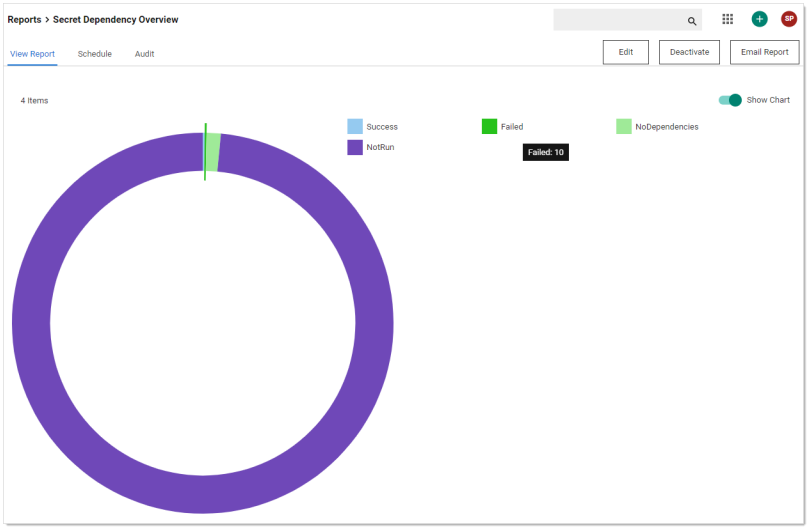
Secret Dependency Not Run

Figure: Secret Dependency Not Run Report



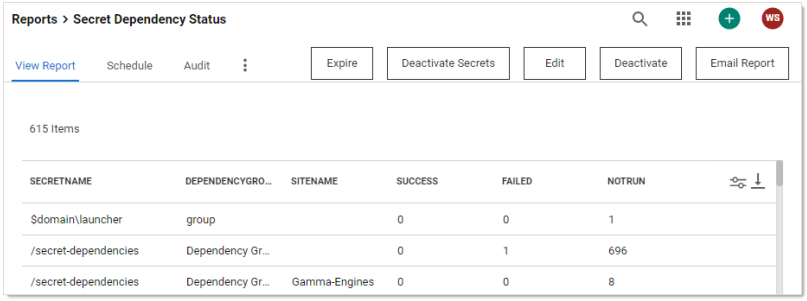
Secret Dependency Overview

Figure: Secret Dependency Overview Report



Secret Dependency Status

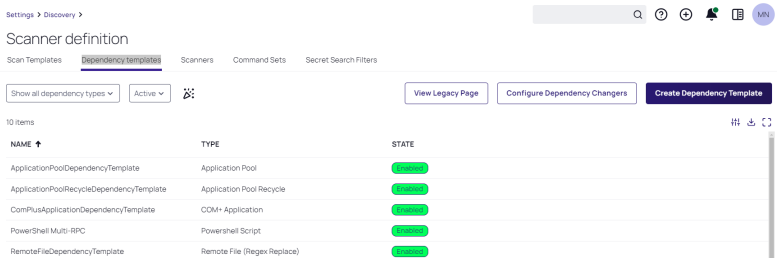
Figure: Secret Dependency Status Report



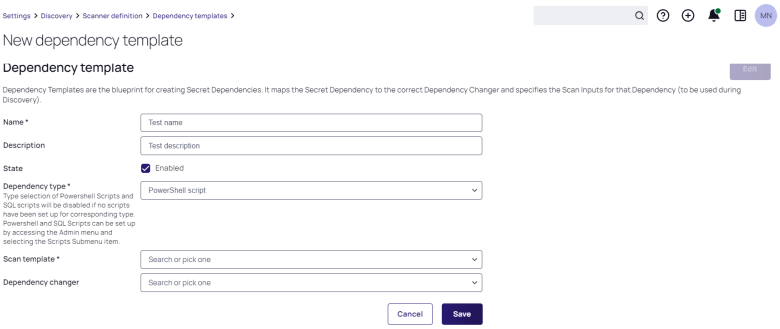
Manually Adding Dependencies

To manually add a dependency:

1. Search for **Admin > Discovery Dependency Templates**. The Scanner definition page appears on the Dependency templates tab:



2. Click **Create Dependency Templates**. The New dependency template page appears:



3. Choose your dependency type from the **Type** dropdown list. Type selection of PowerShell Scripts and SQL scripts will be disabled if no scripts have been set up for corresponding type. PowerShell and SQL Scripts can be set up by accessing the Admin menu and selecting the Scripts Submenu item.
4. Click to select your desired scan template from the **Scan Template** dropdown list.
5. Click to select your desired dependency changer from the **Dependency Changer** dropdown list.
6. Fill in the dependency name and description.
7. Ensure the **Enabled** check box is selected.
8. Click the **Save** button to finish adding the dependency.

Using Regex with Dependencies

Overview

In release version 7.8.00010 and later, Secret Server allows a secret to have file dependencies. File dependencies allow text files with embedded credentials to be changed via Regex.

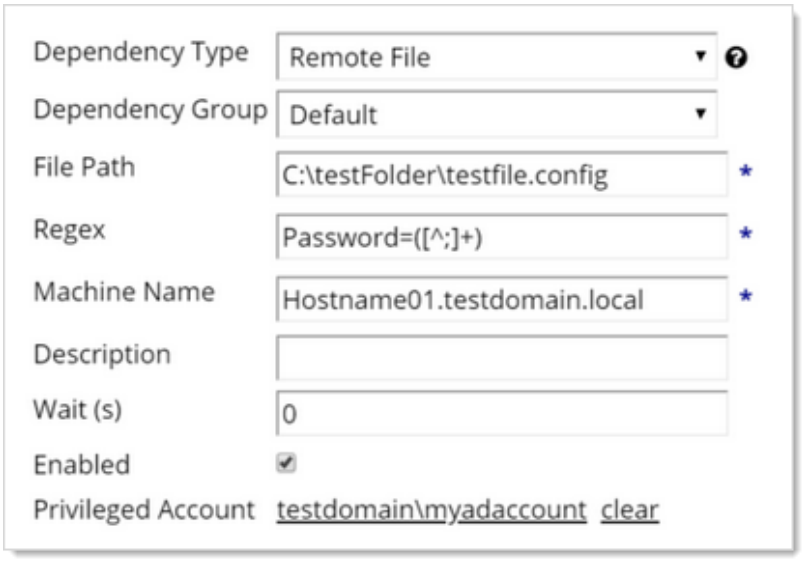
A Regular Expression (Regex) is a phrase in a language for matching text. For details on the .NET Regex language, see .NET Framework Regular Expressions.

Secret Server replaces the contents of the first group (within parentheses) within the Regex.

Setting up a remote file dependency, requires:

- **File Path:** This is the file path on the remote server where the remote password exists. UNC paths do not work here. See UNC Names.
- **Regex:** This regular expression to be used to locate the password embedded in the configuration file.
- **Machine Name:** Computer name or IP address where the dependency is located.
- **Privileged Account:** The account Secret Server will authenticate as when changing the dependency. It must have privileges on the remote machine.

A typical filled in New Dependency page looks something like this:



UNC Names

UNC names, such as:

`\\BARAKA\SHARE\test.txt` or

`\\192.168.1.154\SHARE\test.txt`

do **not** work in the file path. You can, however, put the machine name or IP address in the Machine Name text box, and put the rest of the path in the file path. For example:

In the **File Path** text box:

`\SHARE\test.txt` or

`SHARE\test.text`

In the **Machine Name** text box:

`192.168.1.154` or

`BARAKA`

Examples

The following are some examples of using Regex within file dependencies:

XML Configuration Files

Example One

Source

```
<Configuration>
  <User>
    <UserName>Bob</UserName>
    <Password>Password1</Password>
  </User>
  <User>
    <UserName>Sam</UserName>
    <Password>DontChangeThisOne</Password>
  </User>
</Configuration>
```

Regex

```
<UserName>Bob</UserName>\s*<Password>([^<]+)</Password>
```

Example Two

Source

```
<Configuration>
  <User name="Bob" password="Password1" />
  <User name="John" password="Password1" />
</Configuration>
```

Regex

```
<User name="Bob" password="([^"]+)" />
```

Windows Initialization (.ini) Files

Source

```
[owner]
name=John Doe
password=Password1
organization=Acme Widgets Inc.
```

Regex

```
name=John\sDoe\s*password=([^\r\n]+)
```

SQL Server Connection Strings

Source

```
Data
Source=myServerAddress;Initial Catalog=myDataBase;UserId=myUsername;Password=
myPassword;Server=myServerAddress;Database=myDataBase;Trusted_Connection=False;
```

Regex

```
Password=([^;]+)
```

Oracle Connection Strings

Example One

Source

```
Data Source=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=MyHost)(PORT=MyPort)))
(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=MyOracleSID)));
User Id=myUsername;Password=myPassword;
```

Regex

```
Password=([^;]+)
```

Example Two

Source

```
Data Source=username/password@//myserver:1521/[my.service.com](http://my.service.com);
```

Regex

```
username/([^@/]+)
```

YAML

Source

```
receipt: Oz-Ware Purchase Invoice
date: 2007-08-06
user:
name: Dorothy
password: Password1
```

Regex

```
name:\s*Dorothy\s*password:\s*([^\r\n]+)
```

RPC Errors

Remote Password Changing (RPC) errors in Secret Server can arise from various issues related to configuration, permissions, and connectivity. Common errors include user name not found, access denied, invalid password, and account locked out. These errors often stem from incorrect machine names, firewall settings, domain policy restrictions, or insufficient permissions. For instance, the "ERROR_CANT_ACCESS_DOMAIN_INFO" error may indicate a problem with machine name resolution or domain permissions, while "NERR_PasswordPolicySettings" typically points to password policy violations such as minimum password age. Troubleshooting these errors involves verifying the configuration settings, ensuring proper network connectivity, and checking the relevant logs for detailed error messages.

Common RPC Errors

Overview

This topic lists some of the common errors experienced when setting up Remote Password Changing (RPC) for an account.

To view the errors, navigate to **Admin > Remote Password Changing** in Secret Server and look for the name of the secret you are testing.

Errors

The user name cannot be found

For local Windows accounts, ensure you only have the username in the username field (do not include the machine name). The machine name should go in the Machine field only. If the RDP Launcher stops working when you remove the machine name from the username field, see "RDP Proxy Configuration" on page 817.

Change password failed: Unknown. (ERROR_CANT_ACCESS_DOMAIN_INFO)

For RPC on local Windows accounts, this error can be deceptive because the built-in Windows method used to change a password takes either a machine or domain name, so if the machine is not found, it thinks a domain was passed in and throw a domain error.

For RPC on Active Directory accounts, this error may occur if the account does not have permission to perform the password change or the domain name is wrong or abbreviated. Verify by checking the account properties in Active Directory or log in to the account and try to change the password manually or use privileged secret to perform the RPC.

> The RPC process uses information from the secret, not a central configuration for resetting the password. The Active Directory configuration settings are used for user synchronization only, so ensure the information on the secret is correct, including the Active Directory domain.

Common causes include:

- The machine name is wrong or abbreviated. For example: comp3 is entered as the machine name instead of comp3.yourdomain.com.

  Try replacing the machine name in the secret with the IP address of the machine and seeing if you still receive the domain error.
- The firewall is blocking the ports. See "Ports and IP Addresses Used by Secret Server" on page 765.

Monitor activity to see if the authentication is accepted on the machine by viewing the security log:

1. Run secpol.msc from the Run prompt.
2. Click on Local Policies, Audit Policy.
3. Turn on "Audit account logon events" and "Audit logon events" for both Successes and Failures.
4. View the logs at Administrative Tools > Event Viewer. Check the Security Logs to determine whether the requests are getting through to the computer.

   > The RPC log looks different if the firewall denies the connection, and will show ERROR_ACCESS_DENIED in some cases.

   Firewall settings also apply to changing passwords on the local machine that Secret Server is running on because net authentication is used.

Change password failed: Unknown. (NERR_PasswordPolicySettings)

Cause: repeating password or a password that does not meet domain standards.

Check the minimum password age. When performing RPC on Active Directory accounts, this error may occur due to a minimum password age policy on the domain. If the minimum password age is set to 1 day or greater, and due to testing, the password has already been changed once, a follow up password change will violate the domain policy.

If you need to change the user's password more than the policy allows, change their policy so they are not subject to minimum password ages, or use the privileged account option in the Remote Password Changing tab on the secret. Privileged account will perform a password reset instead of changing the password using the accounts credentials.

Change password failed: Unknown. (ERROR_ACCESS_DENIED)

Cause: User account is set to Not Able to Change Password, firewall denial, or login incorrect. May also occur when the userWorkstations attribute on the user is set.

Change password failed: Unknown. (ERROR_INVALID_PASSWORD)

Cause: Either the user does not exist (ensure the usernames match) or the password is not correct.

Change password failed: Unknown. (ERROR_ACCOUNT_LOCKED_OUT)

Cause: User account is locked out.

ExpiredSecretMonitor - Unspecified error

Cause: Firewall issue or ports are blocked.

DirectoryEntry.Invoke SetPassword - The network path was not found.

Cause: Domain cannot be found from the computer. Check the machine can ping the domain.

Secret '[secretname]' (Id = [secretid]) returned (ArgumentError). Exception: Command requires associated Item. ($[1]$USERNAME)

Cause: the password changer for the secret template this secret is based on is looking for an "associated secret." Associated secrets are additional accounts that are needed in the password change process.

You can view the commands being used for the password change and add the associated Secret by going to the Remote Password Changing tab of the Secret in question and clicking Edit (you may also need to click Show Commands).

Error changing password - Check Out is enabled on associated Secret.

Cause: The secret has a Privileged Account Credentials option selected for performing the password change and the privileged account secret has the Require Check Out option enabled. This configuration causes an error with the remote password change process because the Required Check Out option is not intended for use by the system to avoid conflict from user's request, which is the intended usage.

RPC Error Codes

The most common RPC errors are:

- **NERR_PasswordPolicySettings**: The password Secret Server attempted to set is a repeating password or one that does not meet domain password policy standards. A common reason is minimum password age, which is often defaulted to 24 hours.
- **ERROR_ACCESS_DENIED**: The user account's "Not Able to Change Password" setting could not be set or logon was denied.
- **ERROR_INVALID_PASSWORD**: Either the user does not exist (ensure the usernames match) or the password is not correct.

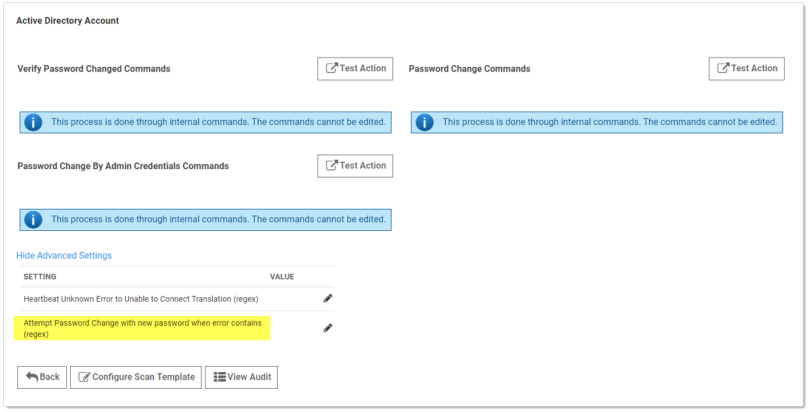
For more information about common error messages for Remote Password Changing, see "Common RPC Errors" on page 946.

Triggering an RPC When Defined Errors Occur

When the "Attempt Password Change with new password when error contains (regex)" setting is enabled, Secret Server generates a new password to use during the next RPC attempt when the defined error is returned. Using a regular expression, which you define, Secret Server scans the error message for specific text strings. When there is a match, Secret Server generates and sets a new next password for the secret that will be used in the next RPC attempt, which will occur based on the templates RPC interval. To keep this process from generating too many next passwords, it is restricted to five attempts while failing RPC.

Only the password field is updated. Passcodes and SSH keys are left alone.

Figure: Attempt Password Change with new password when error contains (regex) setting



Logic:

(RPC Error) AND (one or more regex matches) AND (five or fewer attempts) => New password generated

Examples:

`.*UnknownError.*` (any unknown error)

`.*` (any error)

`.*minimum.*` (minimum password length requirement error)

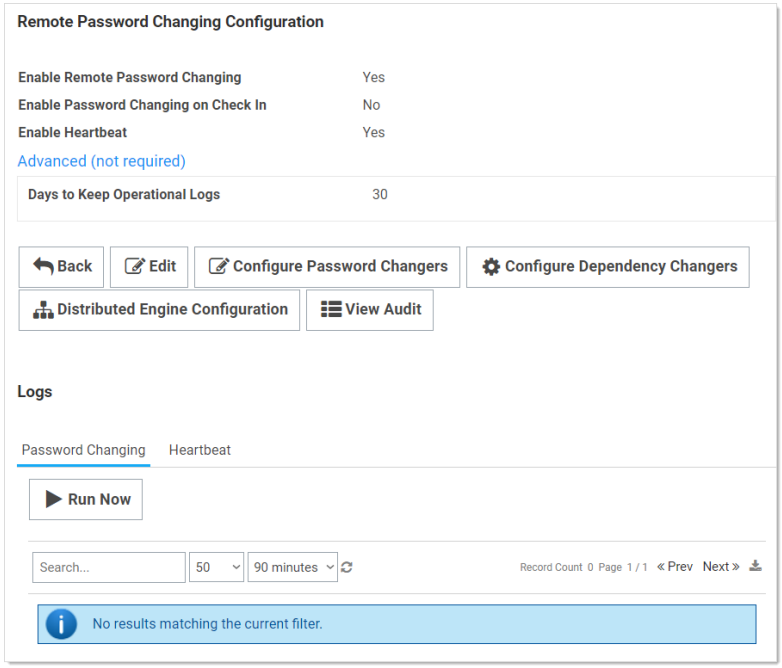
`.*0x80072035.*` (server rejects password error)

`.*0x80072035.*|.*minimum.*` (server rejects password or password length error)

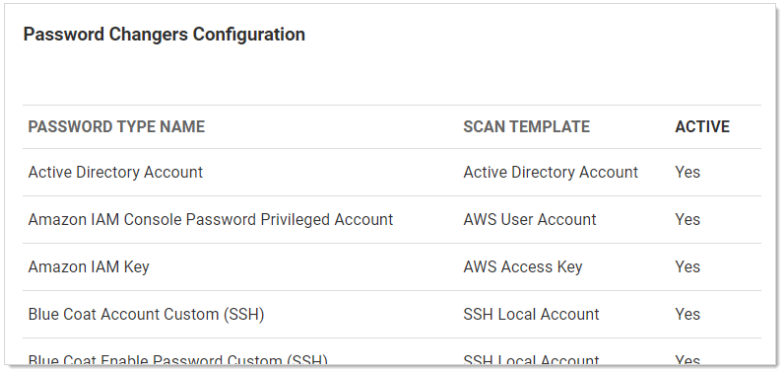
Procedure

To configure RPC in response to specific unknown errors:

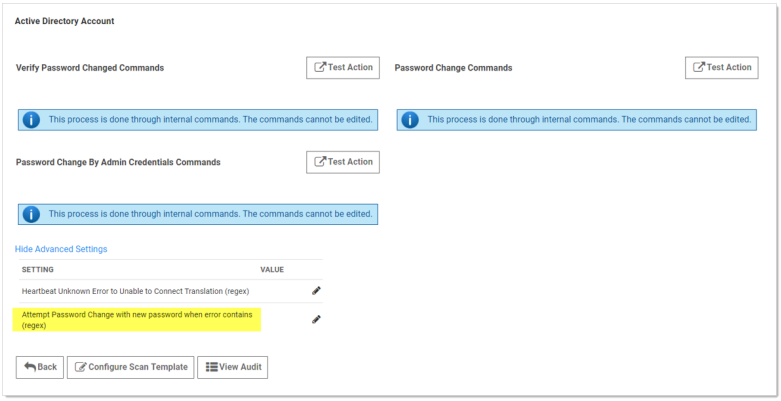
1. Go to **Admin > Remote Password Changing**. The Remote Password Changing Configuration page appears:



2. Click the **Configure Password Changers** button. The Password Changers Configuration page appears:



3. Click the link for the desired password type. Its Account page appears:



4. If necessary, click the **Advanced Settings** link.
5. Click the pencil icon next to **Attempt Password Change with new password when error contains (regex)**. The Value text box appears.
6. Determine the desired text string to search for.
7. Type the desired regex in the **Value** text box.
8. Click the **Save** icon next to the text box.

Viewing RPC Logs

The RPC logs for a specific secret can be accessed by clicking the **View Audit** button on Secret View page and ticking the check box at the bottom of the page for display password changing Log. The RPC logs for all secrets can be accessed by navigating to **Admin > Remote Password Changing**.

You can change the **Days to Keep Operational Logs** text box to set the period to keep RPC-related logs that might contain PII. Secret Server automatically deletes logs older than that (in days).

Included RPC Templates

Delinea Secret Server's out-of-the-box secret templates for remote password changing are designed to automate and secure the process of changing passwords across various systems and accounts. These templates provide a customizable framework, enabling organizations to define specific criteria and procedures for password updates, ensuring that password changes are executed consistently and securely, without manual intervention, thereby enhancing overall security and compliance.

> This section of topics is about using the provided templates, not about configuring RPC to support a given template. For that, see "RPC for Specific Vendors and Technologies" on page 991.

Amazon IAM Console Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Amazon IAM Console accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords

when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Configuring Amazon IAM

Secret Server can scan Amazon Web Services (AWS) for accounts that can access the cloud resource. Secrets based on the "AWS Console Account" secret template can be discovered and managed through the Secret Server.

> The Secret Server heartbeat feature is not available with this template due to AWS IAM limitations.

You can change passwords and access unique passwords for the password secrets in the Amazon IAM console. An Amazon IAM Key should be connected to an "Amazon IAM Console Password" secret to enable password modification. For details, see "Password Management in AWS" on page 604.

Assigning a Password Changer to a Secret Template

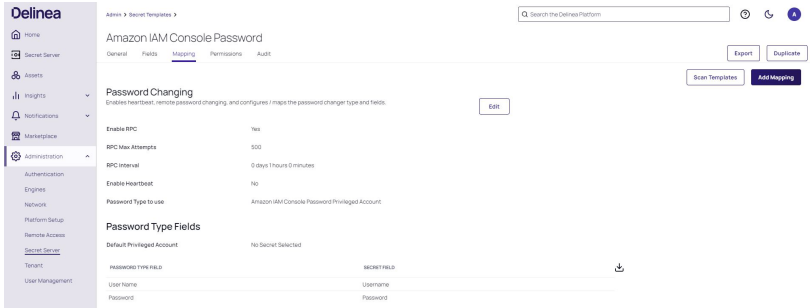
Remote Password Changing (RPC) provides pre-configured password changers assigned to specific secret templates available out of the box. You can view and modify secret templates in the Secret Server administration panel. All possible modification options for secret templates are described in "Creating or Editing Secret Templates" on page 1171. Ensure that the secret template is active. For details, see "Activating and Deactivating Templates" on page 1160.

To navigate to an Amazon IAM Console secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select an Amazon IAM Console secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that the Amazon IAM Console Password Privileged Account RPC conforms to the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Amazon IAM Console template on a secret, see "Managing Secrets" on page 1340.



Amazon IAM Key Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Amazon IAM Key accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Secret Server can scan Amazon Web Services (AWS) for accounts that can access the cloud resource. The secrets based on the "Amazon IAM Key" templates can be discovered and managed through the Secret Server.

An Amazon IAM key secret should be connected to an Amazon IAM console password secret to enable password modification. For details, see "Password Management in AWS" on page 604.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Amazon IAM Key, we want the Amazon IAM Key template.

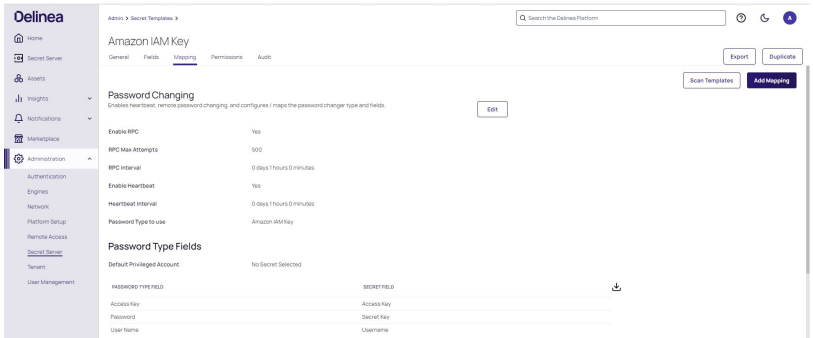
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to an Amazon IAM secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select an Amazon AIM secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that the Amazon IAM Key RPC conforms to the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To utilize the Amazon IAM Key template on a secret, see "Managing Secrets" on page 1340.



Azure Active Directory Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Azure AD accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Configuring Azure AD

To use Secret Server for Azure AD account scanning, we currently rely on an Active Directory scan. For details, see "Setting Permissions for Active Directory Scans" on page 594.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Azure AD, we want the Azure AD template.

You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to an Azure Active Directory secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select an Azure Active Directory secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The Azure AD RPC conforms to the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Azure Active Directory template on a secret, see "Managing Secrets" on page 1340 documentation.

Cisco Account (SSH) Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Cisco Account (SSH) and Cisco Account (Telnet) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Secret Server can use scripted password changers for devices that support SSH or Telnet (this allows for flexibility in changing passwords on less common devices). You can also run custom RPC PowerShell scripts to conduct password changes to other platforms.

With the help of the Secret Server, admins can use private SSH keys for PuTTY launcher sessions and RPC tasks–configurable through password changer settings. Secret Server supports SSH key rotation on secrets. For more details, see "Secret Key Rotation" on page 1454.

Assigning a Password Changer to a Secret Template

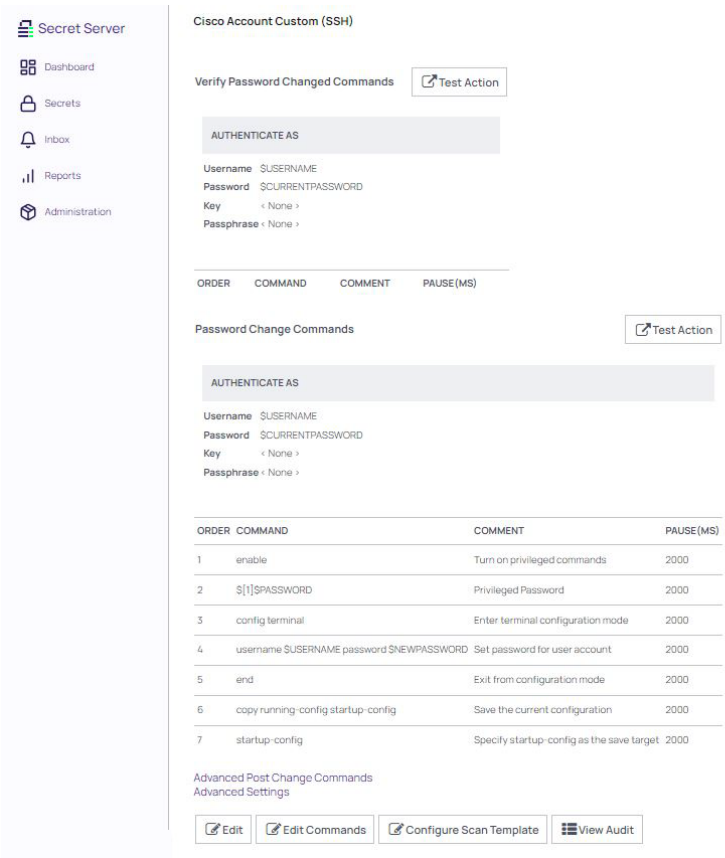
After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Cisco Account, we want the Cisco Account template.

You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a Cisco Account secret template:

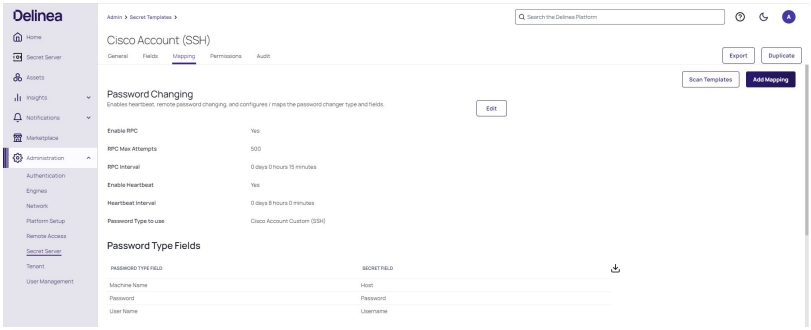
1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a Cisco Account secret template and then click the **Mapping** tab.

It is available to edit custom password changer commands. For more details, see "Editing Custom Commands" on page 1032.



You can check what secret template applies to the selected RPC. The screenshot below shows that a Cisco Account Custom (SSH) RPC is based on the Cisco Account (SSH) secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret Templates determine the fields, launchers, and the remote password changer for secrets. To utilize the Cisco Account template on a secret, see "Managing Secrets" on page 1340 documentation.



Cisco Enable Secret (Telnet) Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Cisco Enable Secret (SSH) and Cisco Enable Secret (Telnet) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Secret Server can use scripted password changers for devices that support SSH or Telnet (this allows for flexibility in changing passwords on less common devices). You can also run custom RPC PowerShell scripts to conduct password changes to other platforms.

With the help of the Secret Server, admins can use private SSH keys for PuTTY launcher sessions and RPC tasks–configurable through password changer settings. Starting with version 10.1.000000, the Secret Server supports SSH key rotation on secrets. For more details, see "Secret Key Rotation" on page 1454.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Cisco Enable Secret (SSH), we want the Cisco Enable Secret (SSH) template.

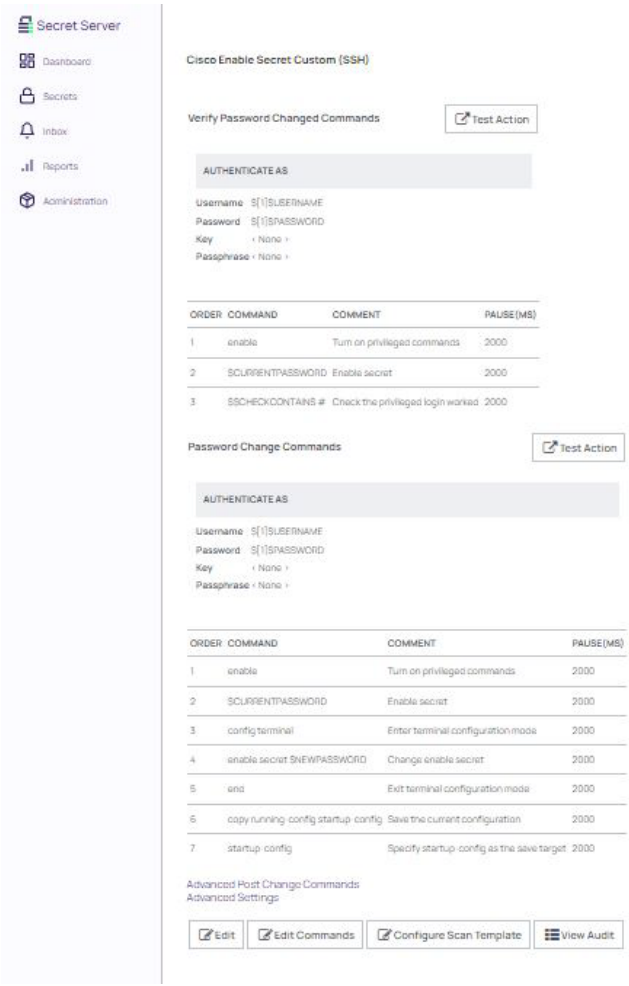
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a Cisco Enable secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.

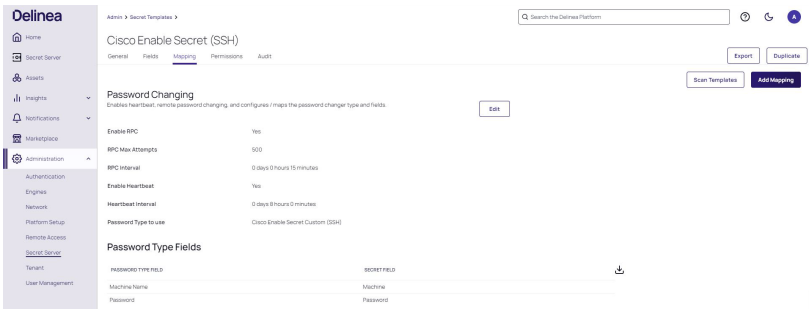
3. Select a Cisco Enable secret template and then click the **Mapping** tab.

It is available to edit custom password changer commands. For more details, see "Editing Custom Commands" on page 1032.



You can check what secret template conforms to the selected RPC. The screenshot below shows that a Cisco Enable Secret Custom (SSH) RPC refers to a Cisco Enable Secret (SSH) template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Cisco Enable Directory template on a secret, see "Managing Secrets" on page 1340.



Entra ID Secret Template for RPC

Overview

> For setting up the password changer itself, see "Configuring an Azure AD or Entra ID Password Changer" on page 1002

Introduction

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Entra ID accounts. With RPC secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the Password Changer List for a complete list of available password changers.

Secret Server has supported Azure AD remote password changing for several years, this overhaul creates a new password changer and template which use Oauth application credentials as a privileged account, to change a user password.

> In July 2023, Microsoft rebranded Azure AD to Microsoft Entra ID to improve consistency with its other Entra cloud products.

Entra ID

Entra ID is Microsoft's cloud-based identity and access management (IAM) solution. Key points about Entra ID:

- It is a directory and identity management service that operates in the cloud. It provides authentication and authorization services for various Microsoft services like Microsoft 365, Dynamics 365, and Microsoft Azure.
- Entra ID enables a single sign-on experience for users, regardless of whether their applications are cloud-based or on-premises.
- It offers multiple authentication methods including password-based, multi-factor, smart card, and certificate-based authentication.
- Entra ID includes security features like Conditional Access policies, risk-based authentication, and identity protection.
- Entra ID provides benefits to different members of an organization based on their roles. This can include giving

IT admins control over app access, enabling developers to easily integrate single sign-on, and providing a unified identity management solution for Microsoft 365, Azure, and Dynamics 365 subscribers.

In summary, Entra ID is Microsoft's comprehensive cloud-based identity and access management solution that helps organizations securely manage identities and access across their Microsoft services and applications.

Template Benefits

- Supports MFA
- Does not require PowerShell

Creating an Entra ID Secret and Assigning it to Individual-Account Secrets

> The following steps are meant for users who either utilize an App Registration created during Discovery or already know how to create a separate App Registration in Entra ID. This is a prerequisite to proceed and If you do not have this set up, see "Entra ID Discovery" on page 607 to create and configure an App Registration in Entra ID.

To create a privileged secret and assign it to individual account secrets:

1. Access **Secrets > All secrets** and create a secret of the **Azure Application Registration** type.
2. Type in values for these fields:
   - Secret name
   - Client ID
   - Client Secret
   - Tenant ID
3. Click **Create secret**.
4. For each Entra ID user that corresponds to the above credentials, create a secret of the **Entra ID User Account** type.
5. Type in values for the following fields:
   - Secret name
   - Username
   - Password
   - Domain and Notes are optional fields
6. For Entra ID user accounts, heartbeat and password changing will only work if a privileged account of the **Azure Application Registration Account** type is set:
   a. Click the **Remote Password Changing** tab for the secret you just created.
   b. Click **Edit** under **RPC / Autochange**.
   c. Select the **Privileged Account Credentials** radio button for the **Change Password Using** option.
   d. Click **No Secret Selected**. A list of eligible Secrets appears.

e. Select a secret with credentials that would have permissions over the user account. If no secrets appear, check that there are secrets of the **Azure Application Registration** type that are visible to the logged in user.

> Until a privileged account is set, heartbeat and RPC options are unavailable and will fail if running automatically.

Generic Discovery Credentials Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Generic Discovery Credentials accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Generic, we want the Generic Discovery Credentials template.

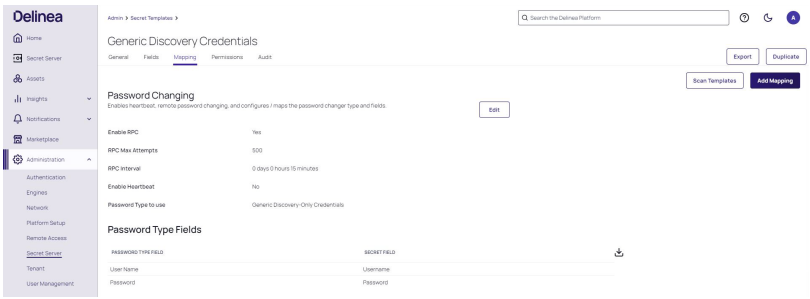
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a Generic Discovery Credentials secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a Generic Discovery Credentials secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows the Generic Discovery-Only Credentials RPC refers to the Generic Discovery Credentials secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Generic Discovery Credentials template on a secret, see "Managing Secrets" on page 1340.



Google IAM Service Account Key Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Google IAM accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Secret Server can manage Google Cloud Platform (GCP) service accounts and VM instances. For more details, see "Google Cloud Platform Discovery" on page 613.

Assigning a Password Changer to a Secret Template

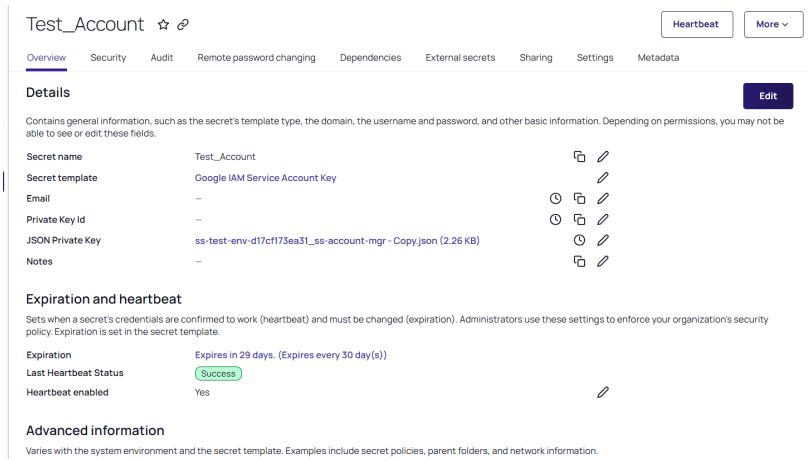
After completing the RPC setup, you can manage the built-in secret templates. Each secret template is a specific application and is preconfigured with the password changer best suited to that. For the Google Cloud Platform, we want the Google IAM Service Account Key template.

You can view and modify secret Templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

Navigate to **Admin > Secret Templates**, and select Google IAM Service Account Key template from the list (or use the search box if you don't see it). On the template page select the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows the Google IAM Service Account Key RPC refers to the selected Google IAM secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Google IAM Directory template on a secret, see "Managing Secrets" on page 1340.



Note that for the Google IAM Service Account Key secret template, the Email and Private Key ID fields are not required, as the JSON private key file contains all the necessary information.

HP iLO Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for HP Integrated Lights-Out (iLO) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

HP Integrated Lights-Out (iLO) is an embedded server management technology exclusive to Hewlett Packard Enterprise (HPE) servers, providing a comprehensive suite of tools to ensure seamless server setup, health monitoring, power and thermal control, and remote server administration.

Secret Server can set up scripted password changers for devices that support SSH or Telnet (this allows for flexibility in changing passwords on less common devices). You can also run custom RPC PowerShell scripts to conduct password changes to other platforms.

With the help of the Secret Server, admins can use private SSH keys for PuTTY launcher sessions and RPC tasks–configurable through password changer settings. Starting with version 10.1.000000, the Secret Server supports SSH key rotation on secrets. For more details, see "Secret Key Rotation" on page 1454.

Assigning a Password Changer to a Secret Template

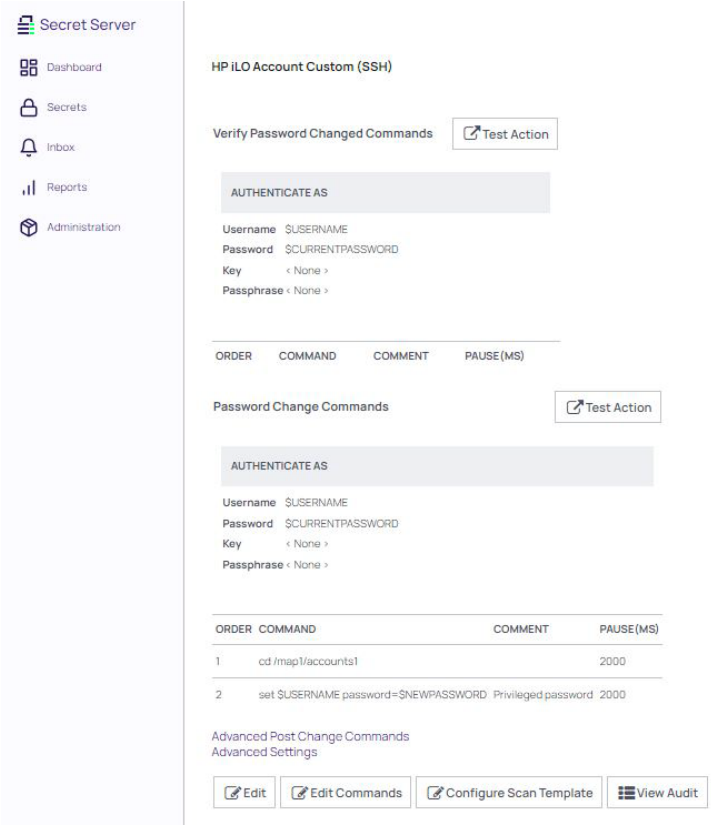
After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the HP iLO, we want the HP iLO template.

You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to an HP iLO secret template:

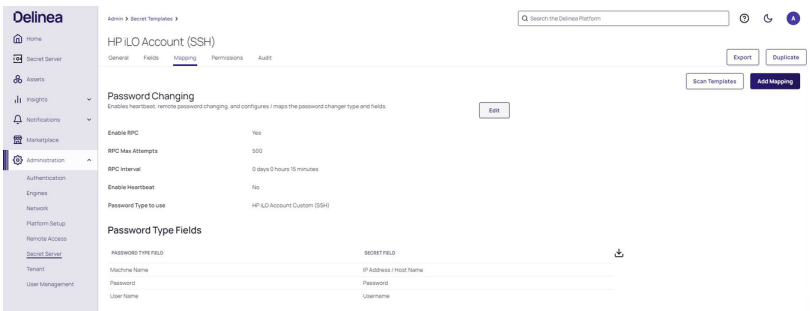
1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select an HP iLO secret template and then click the **Mapping** tab.

It is available to edit custom password changer commands. For more details, see "Editing Custom Commands" on page 1032.



You can check what secret template conforms to the selected RPC. The screenshot below shows that the HP iLO Account Custom (SSH) RPC refers to an HP iLO Account (SSH) secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the HP iLO template on a secret, see "Managing Secrets" on page 1340.



IBM iSeries (AS/400) Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for IBM iSeries (AS/400) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

The IBM iSeries (AS/400) Terminal password changer uses the same principles as the z/OS Mainframe password changer. The password change and heartbeat execute the 5250 terminal connection and scripting. You can modify the script for any advanced configuration requirements. For more details, see "Create and Customize an IBM iSystem (AS/400) Template to use the new IBM iSeries (AS/400) Password Changer" on page 1195 .

Assigning a Password Changer to a Secret Template

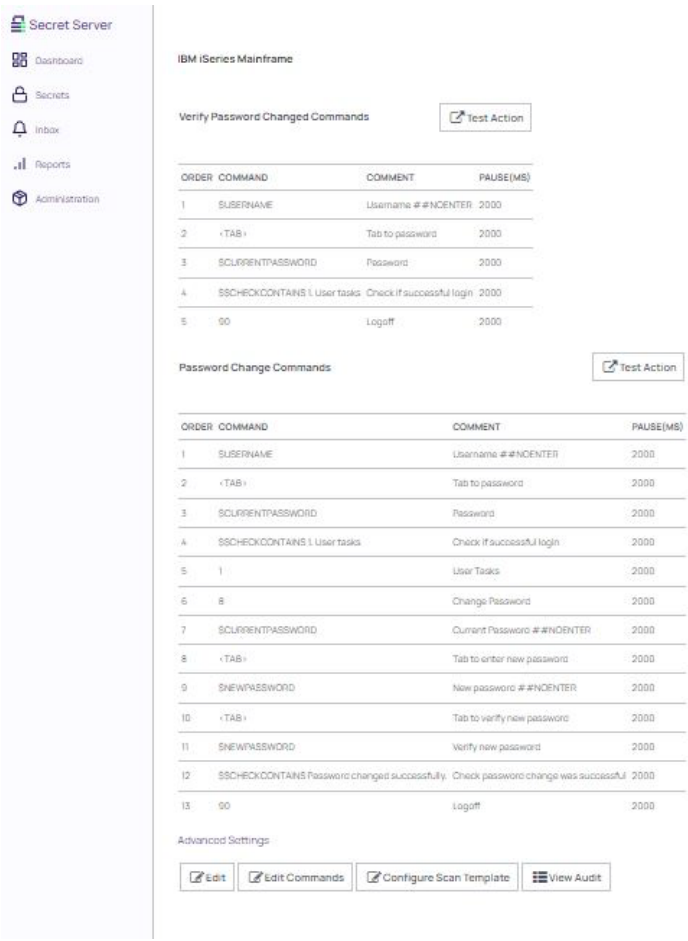
After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the IBM Series (AS/400), we want the IBM Series (AS/400) template.

You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to an IBM Series (AS/400) secret template:

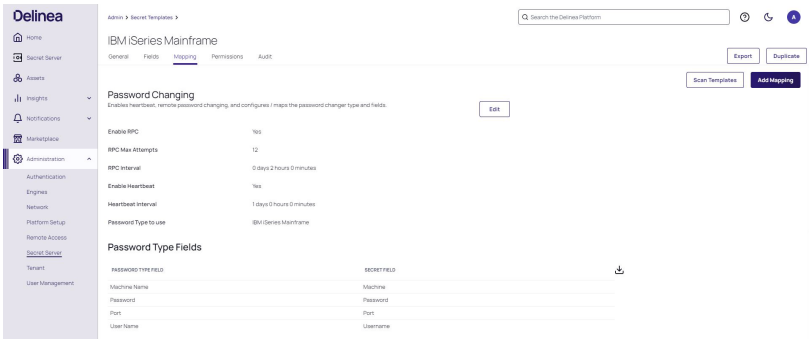
1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select an IBM Series (AS/400) secret template and then click the **Mapping** tab.

It is available to edit custom password changer commands. For more details, see Editing Custom Commands.



You can check what secret template conforms to the selected RPC. The screenshot below shows the IBM Series Mainframe RPC refers to the selected IBM Series (AS/400) secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To utilize the IBM Series (AS/400) template on a secret, see "Managing Secrets" on page 1340.



Microsoft AD Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Microsoft Active Directory (AD) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Prerequisites

RPC setup for AD requires the following:

- Use a privileged account with the correct permissions. Select the Privileged Account Credentials option on an AD secret with permission to change the account's password.
- Ensure the appropriate permissions for Active Directory are enabled. For details, see "Configuring Delegation Control for the Administrative Account" on page 1000.

> The Active Directory password changer has an RPC "timeout minutes" advanced setting. This setting only applies when using the "Password Change By Admin Credentials" option.

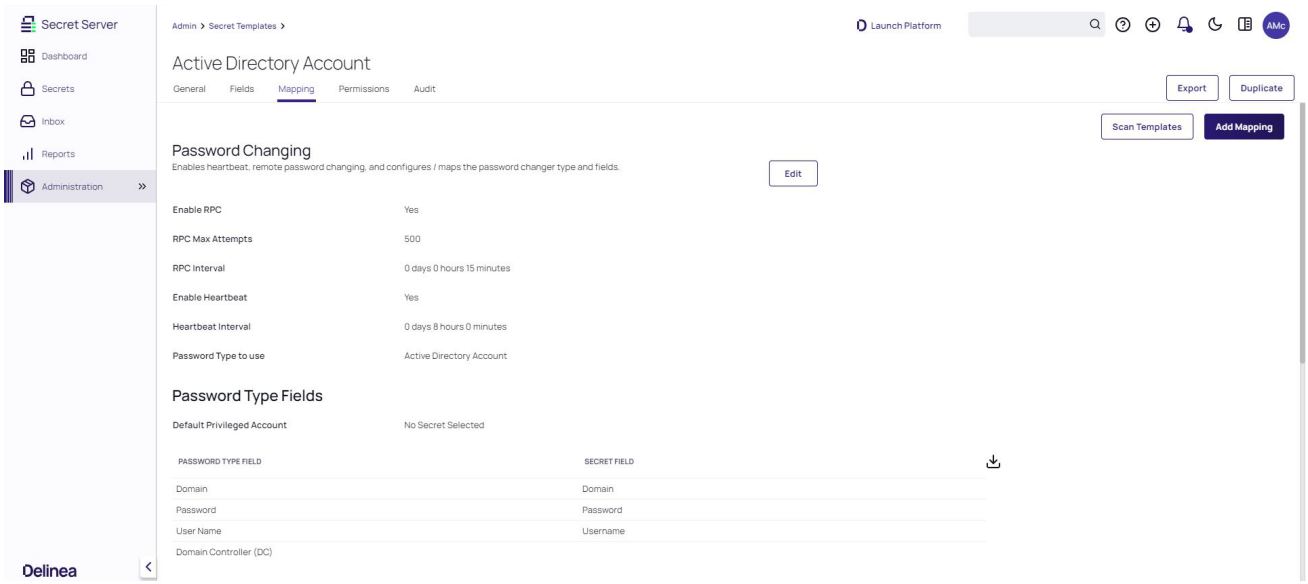
Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For AD, we want the Active Directory Account template.

You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a Microsoft AD secret template:

1. Go to **Administration > Secret Secret Server > Administration**. The Secrets Administration page appears.
2. In the **Core Actions** section, click **Secret Templates**. The Secret Templates page appears with a list of available templates.
3. Click the **Active Directory Account** template name. That template's page appears.
4. Click the **Mapping** tab.
5. You can now view or edit the RPC for the secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.



MySQL Account Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for MySQL Accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

MySQL is a widely-used open-source relational database management system (RDBMS) known for its speed, reliability, and ease of use. It is a central component of the LAMP web application software stack, along with Linux, Apache, and PHP/Perl/Python.

To configure secret templates for MySQL, see "RPC on SQL Server Accounts" on page 1029.

Distributed Engine Considerations

For MySQL RPC to work, a MySQL connector is required on the distributed engine (DE), and the MySQL.data.dll file needs to be added in the DE files. See "Distributed Engine Overview" on page 723 for details.

When DEs auto update, they remove the MySQL, Oracle, and other DLLs that were manually placed there. To forestall this, we recommend creating an ignore file for DE upgrades.

How to Create an Ignore File for Distributed Engine Upgrades

Create and configure an ignore file for Distributed Engine upgrades to allow Distributed Engine ignoring specific DLLs and not replacing them during upgrades.

1. Open a text editor like Notepad or Notepad++.

2. In the content of the file, add the filename of any DLLs that you'd like to ignore during upgrades. If there is more than one file, add each additional filename on a new line.

3. Save this file to the data folder on the Distributed Engine machine. By default, it is: C:\Program Files\Thycotic Software Ltd\Distributed Engine\data.

- Filename must be "ignore" without quotes.
- Uncheck "Append Extension" as this file must not contain an extension.

4. Once complete, any further upgrades will ignore any file listed in the ignore file.

Additionally see video demo for details.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the MySQL Account, we want the MySQL Account template.

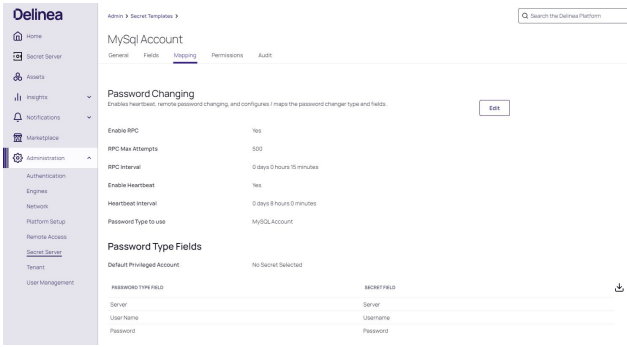
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a MySQL Account secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a MySQL Account secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that the MySQL Account RPC conforms to the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the MySQL Account template on a secret, see "Managing Secrets" on page 1340 documentation.



Okta Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for an Okta Account. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret

expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Okta is a leading identity management platform that enables organizations to securely connect their employees, partners, and customers with the right technology resources. For instructions on how to set up an Okta Account and arrange its secret template, see Remote Password Changing for Okta.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is application specific and is preconfigured with the password changer best suited to it. For Okta, we want the **Okta Account template**.

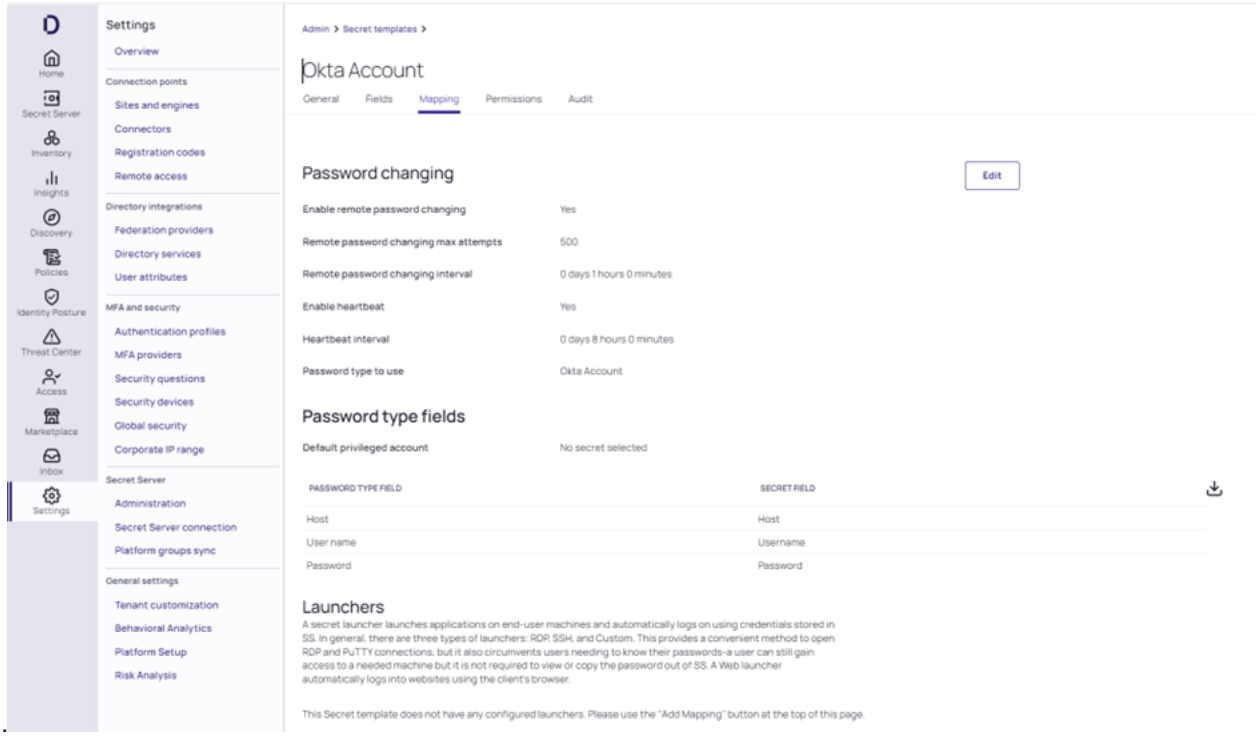
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in the **Active** status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to an Okta Account secret template:

1. Go to **Administration > Secret Server**. The Secrets Administration page will be displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates will be displayed.
3. Select an **Okta Account** secret template and then access the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that the Okta Account RPC conforms to the selected secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To utilize the Okta Account template on a secret, see "Managing Secrets" on page 1340



OpenLDAP Account Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for OpenLDAP Account. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

OpenLDAP is an open-source implementation of the Lightweight Directory Access Protocol (LDAP), widely used for directory services in network environments. It enables the management and access of information about network resources and users, facilitating centralized authentication and authorization.

To use Secret Server for OpenLDAP Account scanning, you should set the integration process. For more details, see"Syncing with OpenLDAP Directory Service" on page 517.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the OpenLDAP, we want the OpenLDAP Account template.

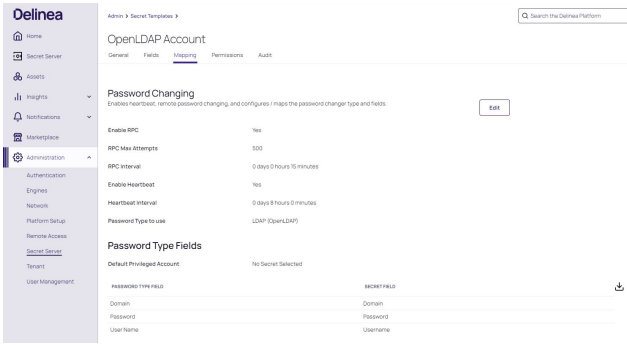
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to an OpenLDAP Account secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select an OpenLDAP Account secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that the LDAP (OpenLDAP) RPC conforms to the OpenLDAP Account secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the OpenLDAP Account template on a secret, see "Managing Secrets" on page 1340 documentation.



Oracle Account Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Oracle Account and Oracle Account (Template Ver 2) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

To configure secret templates for Oracle, see "Configuring Oracle Secret Templates" on page 1205.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Oracle, we want the Oracle Account template.

You can view and modify secret Templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

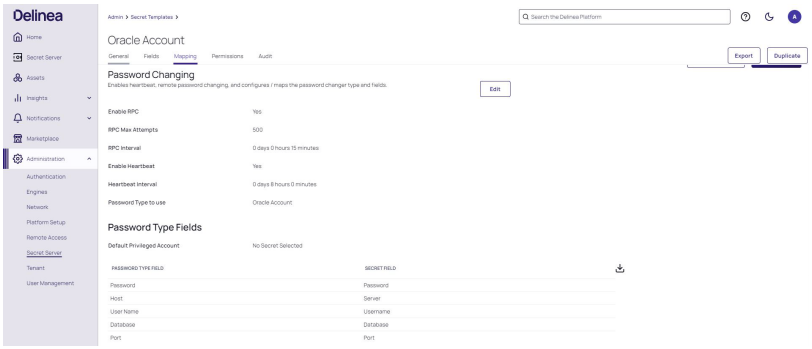
To navigate to an Oracle Account secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.

3. Select an Oracle Account secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that the Oracle Account RPC conforms to the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Oracle Account template on a secret, see "Managing Secrets" on page 1340.



Oracle Account (TCPS) Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Oracle Account (TCPS) and Oracle Account (Walletless) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Oracle TCPS (Transparent Network Substrate over SSL/TLS) is a technology used in Oracle Database environments to provide secure, encrypted communication between clients and servers. It operates as an extension of Oracle's Transparent Network Substrate (TNS), a fundamental part of the Oracle Net Services, facilitating network sessions from clients to Oracle databases.

To configure secret templates for Oracle, see "Configuring Oracle Secret Templates" on page 1205.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Oracle Account (TCPS), we want the Oracle Account (TCPS) template.

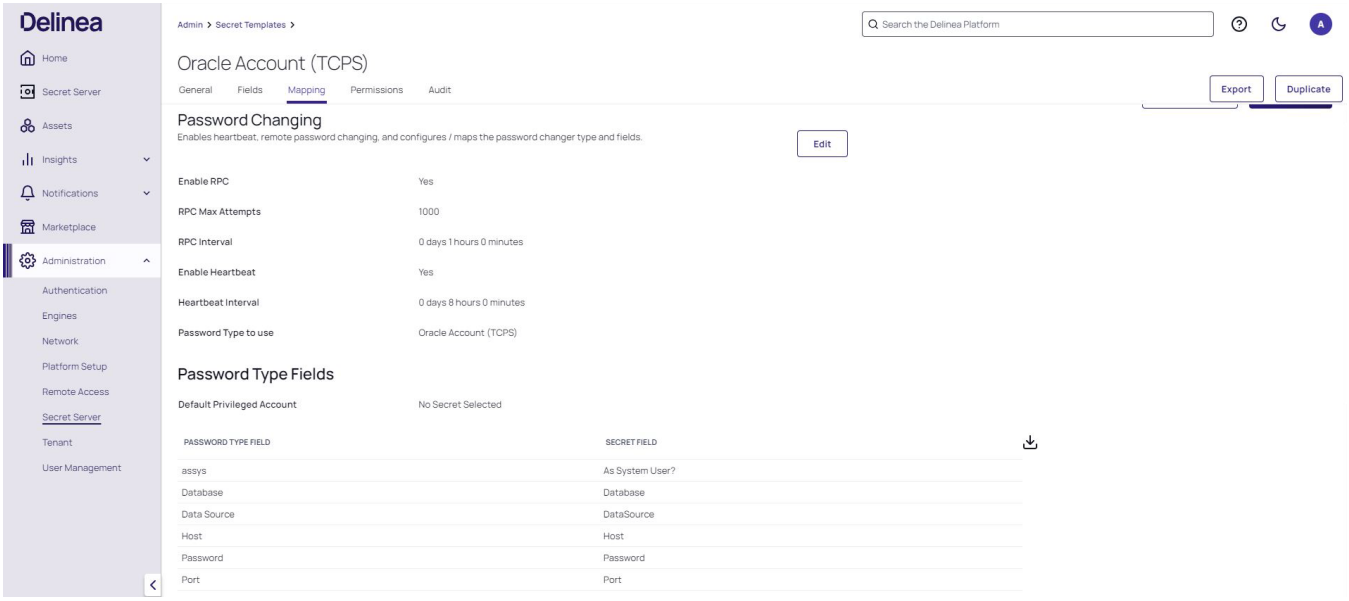
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to an Oracle Account (TCPS) secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select an Oracle Account (TCPS) secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that the Oracle Account (TCPS) RPC conforms to the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Oracle Account (TCPS) template on a secret, see "Managing Secrets" on page 1340.



SAP Account Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for SAP Account. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

SAP, an acronym for Systems, Applications, and Products in Data Processing, is a global leader in enterprise resource planning (ERP) software and related enterprise applications. For instructions on how to set up the SAP Account and arrange its secret template, see "Configuring SAP SNC Account Secret Templates" on page 1184.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the SAP, we want the SAP Account template.

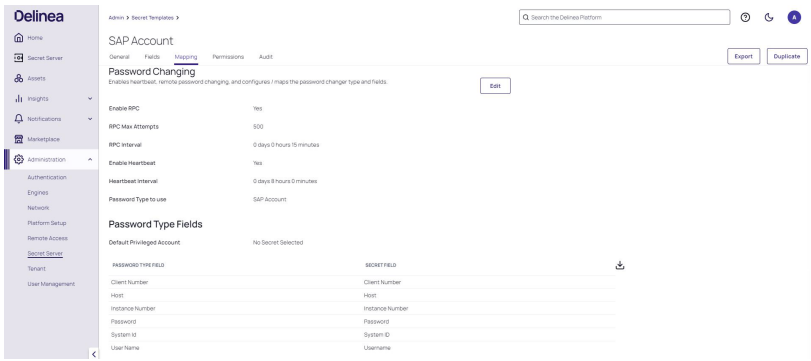
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a SAP Account secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a SAP Account secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that the SAP Account RPC conforms to the selected secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To utilize the SAP Account template on a secret, see "Managing Secrets" on page 1340.



ServiceNow Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for the ServiceNow Account. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

ServiceNow is a software-as-a-service supplier that provides technical management support, such as IT service management, to the IT operations of large corporations, including help desk functionality. For instructions on how to set up the ServiceNow Account and arrange its secret template, see Integrating ServiceNow RPC with Secret Server.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is application specific and is preconfigured with the password changer best suited to it. For ServiceNow, please use the **ServiceNow Account template**.

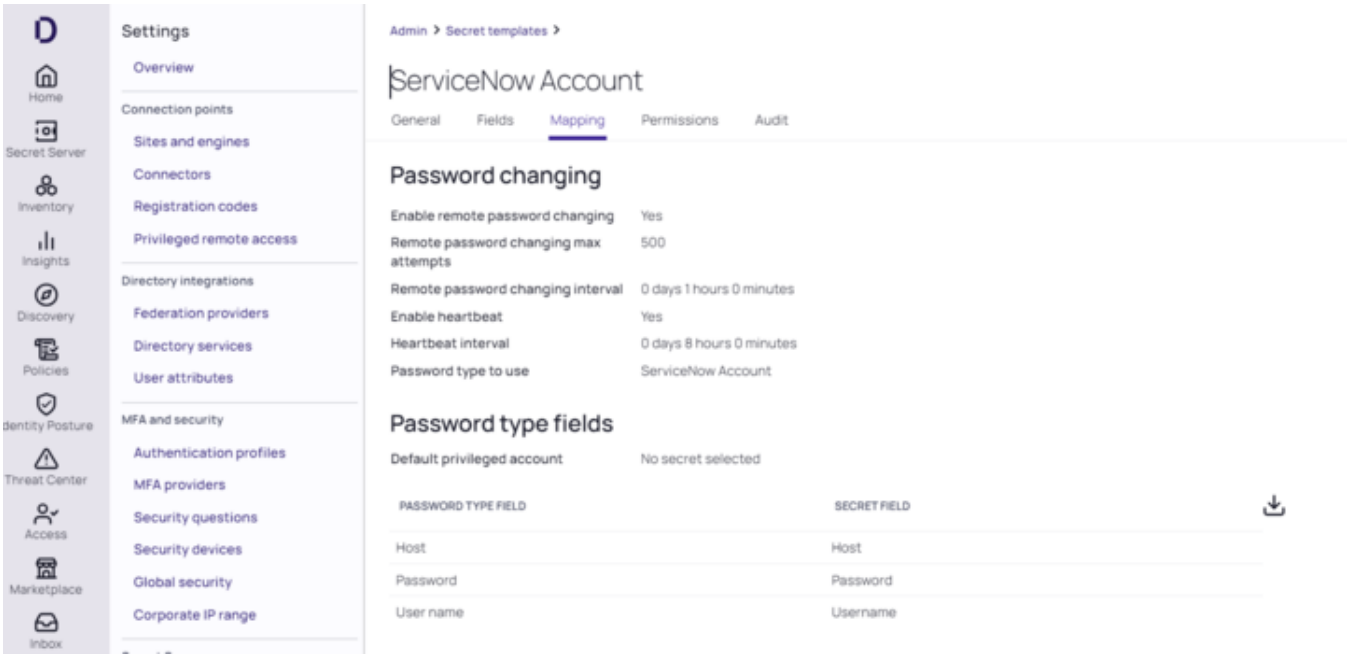
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more details on the available options. Ensure that the secret template is in the **Active** status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a ServiceNow Account secret template:

1. Go to **Administration > Secret Server**. The Secrets Administration page will be displayed.
2. In the **Secrets** section, select **Secret Templates**. The list of available templates will be displayed.
3. Select a **ServiceNow Account** secret template and then access the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that the ServiceNow Account RPC conforms to the selected secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916

Secret templates determine the fields, launchers, and the remote password changer for secrets. To utilize the ServiceNow Account template on a secret, see "Managing Secrets" on page 1340.



RPC for Snowflake in Secret Server

RPC for Snowflake in Secret Server applies to Snowflake SQL database user accounts, including both admin and non-admin user accounts.

Prerequisites

Make sure you have:

- Two active Snowflake accounts. One of these accounts must be a privileged admin account which will be used for password changing.

- A Secret Server user which can create two snowflake secrets.
  - Optionally, admin credentials for Secret Server (the **ACCOUNTADMIN** role must be assigned to the admin account).
- The RPC feature enabled in Secret Server.
- Permission to create and configure secrets.
- Heartbeat monitoring and remote password-changing features enabled on Secret Server.
- A site with a distributed engine which has access to the internet.

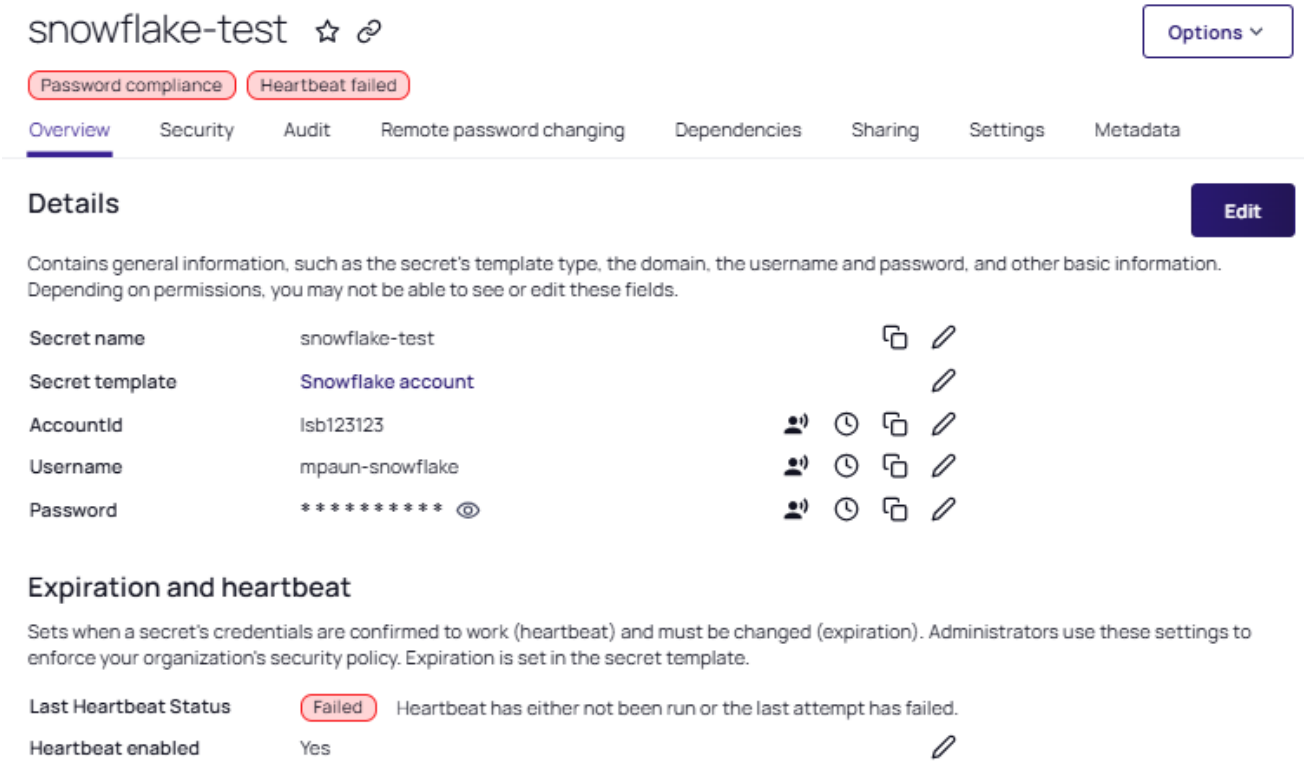
Configuration

1. Log into Secret Server.
2. Navigate to **Secrets >All secrets** and click the **Create secret** button, the **Create new secret** popup appears.
3. Search for the **Snowflake account** template and select it. The popup refreshes automatically to reflect the fields you must fill in.
4. Complete the following fields:
   a. **Secret name**: give the secret an appropriate name.
   b. **AccountId**: you will find this as a part of your Snowflake URL (starts with lsb followed by several numbers).
   c. **Username**: the username used to sign into the Snowflake account.
   d. **Password**: the password used to sign into the Snowflake account.
   e. **Site**: set a site with a distributed engine that can access Snowflake services.
   f. Leave **Auto Change Enabled** unchecked and click **Create secret**. The newly created secret loads automatically for viewing.
5. The **Heartbeat** operation runs automatically to check if the entered credentials are valid. If the credentials are valid the status will change from **Pending** to **Success**.

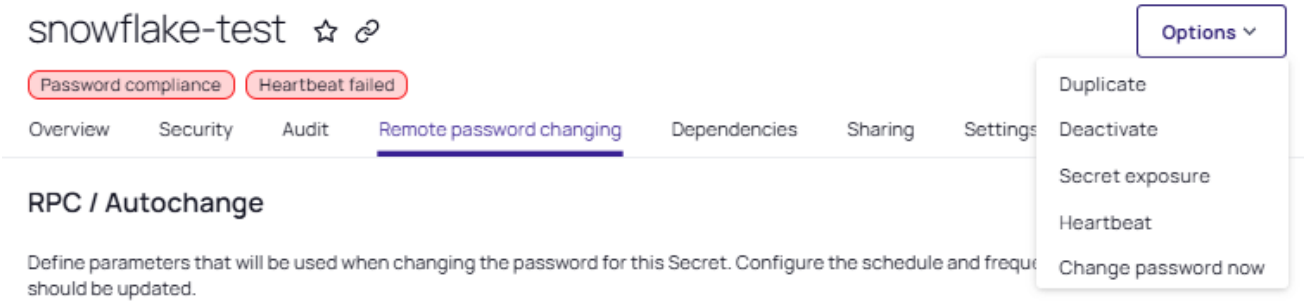
   If the credentials are not valid the status will change from **Pending** to **Failed**.

> The distributed engine checks for RPC every 300 seconds. If the heartbeat state remains in **Pending** for longer than 300 seconds, confirm that the site has an operational distributed engine by accessing **Settings > Sites and engines**.

> To verify the status of the heartbeat processes, navigate to **Settings > Heartbeat Log**.



6. Navigate to the **Remote password changing** tab and select **Edit** for the **RPC/Autochange** section.
7. For **Change password using**, select the **Privileged account credentials** option.
   a. If you chose the option above, the **Change password using** option appears, and you must select a secret by clicking on the **No secret selected** link. A popup will appear where you can search for the secret you want to associate. Select a Snowflake user with the **ADMINACCOUNT** role used to process the password change.
   b. Click **Save**.
8. (Optional) Access the **Change password now** option button from the top right corner if you want to change the secret password. Alternatively, it can be found under the **Options** dropdown list:



SonicWall NSA Web Admin Account Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for SonicWall NSA Web Admin Account. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

SonicWall is a well-established company specializing in network security and data protection solutions. It offers a wide array of products and services designed to safeguard businesses from a variety of cyber threats, including firewalls, VPNs (Virtual Private Networks), anti-spam, and content filtering systems.

The SonicWall Network Security Appliance (NSA) series is a line of advanced firewalls designed by SonicWall, targeted primarily at medium to large-sized businesses and distributed enterprise environments. These appliances provide comprehensive network security through intrusion prevention, anti-malware, content/URL filtering capabilities, and VPN support for secure remote access.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the SonicWall NSA Web Admin Account, we want the SonicWall NSA Web Admin Account template.

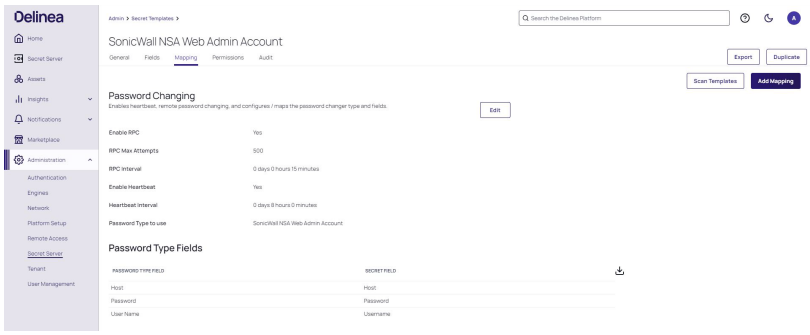
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a SonicWall NSA Web Admin Account secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a SonicWall NSA Web Admin Account secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows the SonicWall NSA Web Admin Account RPC is based on the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the SonicWall NSA Web Admin Account template on a secret, see "Managing Secrets" on page 1340 documentation.



SonicWall NSA Web Local User Account Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for SonicWall NSA Web Local User Account. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the SonicWall NSA Web Local User Account, we want the SonicWall NSA Web Local User Account template.

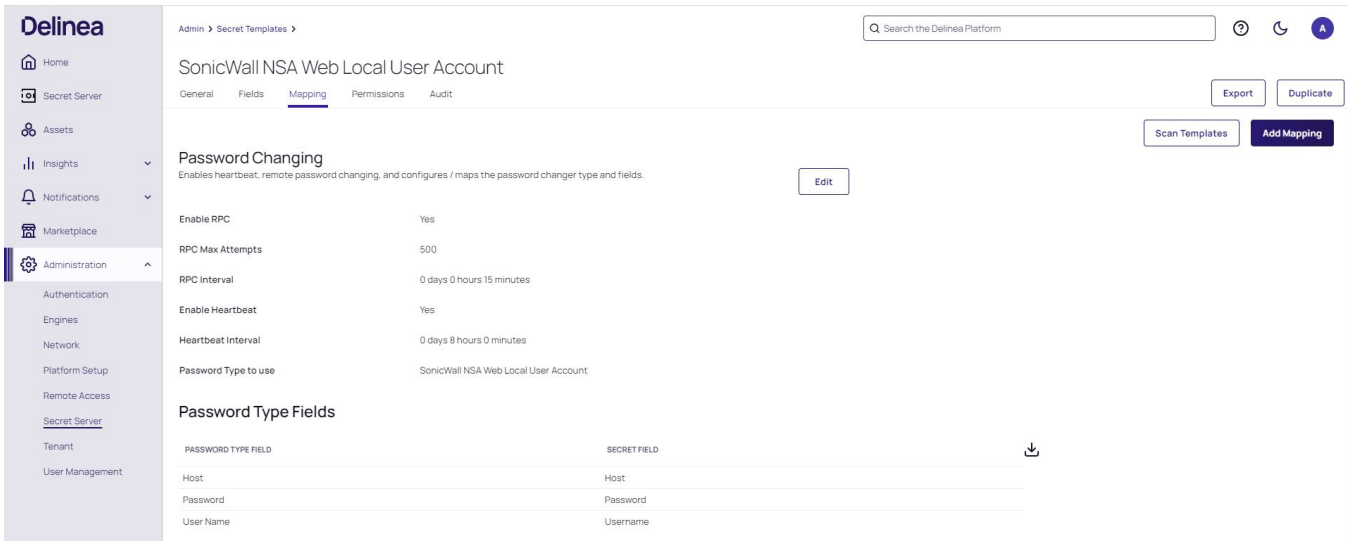
You can view and modify secret Templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a SonicWall NSA Web Local User Account secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a SonicWall NSA Web Local User Account secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows the SonicWall NSA Web Local User Account RPC is based on the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To utilize the SonicWall NSA Web Local User Account template on a secret, see "Managing Secrets" on page 1340.



Sybase Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Sybase accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Sybase SQL Server, was a high-performance relational database management system (RDBMS), which gained significant popularity in the financial industry for its robustness and efficiency in handling large transactions. For more details on Secret Server and Sybase integration, see"Ports and IP Addresses Used by Secret Server" on page 765.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Sybase Account, we want the Sybase Account template.

You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

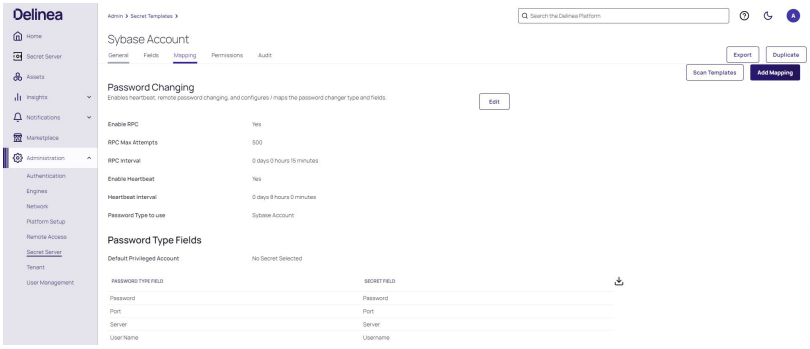
To navigate to a Sybase Account secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a Sybase Account secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows the Sybase Account RPC is based on the identically titled secret template. It is possible to assign several password changers to

one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Sybase Account template on a secret, see "Managing Secrets" on page 1340.



Unix Account (SSH Key Rotation) Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) Unix Account (SSH Key Rotation) and Unix Account (Privileged Account SSH Key Rotation) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

SSH Key Rotation manages a Unix account's private keys, passphrases, and passwords. The public/private key pair is regenerated, and the private key is encrypted with a new passphrase any time a secret's password changes, manually or automatically. The public key is then updated on the Unix machine referenced on the secret. For details, see "Custom SSH Key Rotation" on page 1053.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Unix Account SSH Key Rotation, we want the Unix Account SSH Key Rotation template.

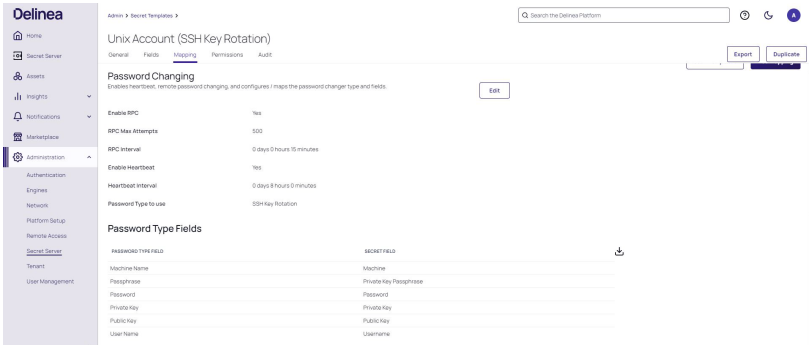
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a Unix Account SSH Key Rotation or Privileged Account SSH Key Rotation secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a Unix SSH Key Rotation or Privileged Account SSH Key Rotation secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that a Unix SSH Key Rotation RPC refers to the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Unix SSH Key Rotation or Privileged Account SSH Key Rotation template on a secret, see "Managing Secrets" on page 1340 documentation.



Unix Account (SSH Key Rotation - No Password) Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Unix Account (SSH Key Rotation - No Password) and Unix Account (Privileged Account SSH Key Rotation - No Password) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

SSH Key Rotation manages a Unix account's private keys, passphrases, and passwords. The public/private key pair is regenerated, and the private key is encrypted with a new passphrase any time a secret's password changes, manually or automatically. The public key is then updated on the Unix machine referenced on the secret. For more details, see "SSH Key Rotation" on page 1049.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Unix Account (SSH Key Rotation - No Password), we want the Unix Account (SSH Key Rotation - No Password) template.

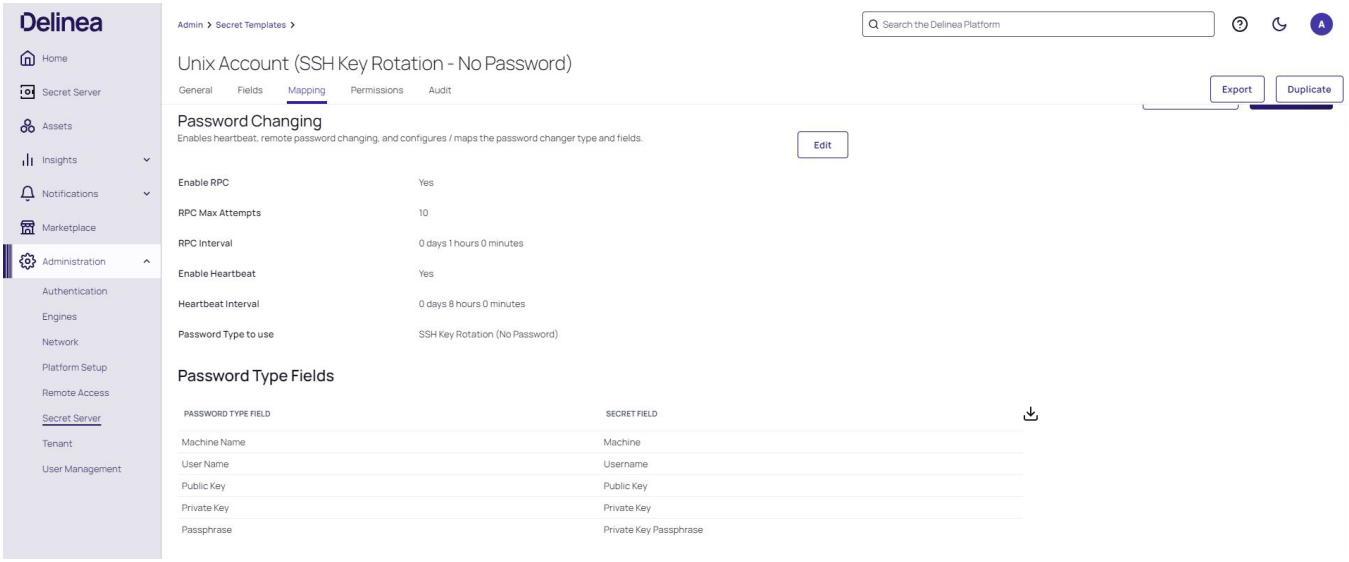
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a Unix Account (SSH Key Rotation - No Password) or Unix Account (Privileged Account SSH Key Rotation - No Password) secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a Unix Account (SSH Key Rotation - No Password) or Unix Account (Privileged Account SSH Key Rotation - No Password) secret template, then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that a Unix Account (SSH Key Rotation - No Password) RPC refers to the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To utilize the Unix Account (SSH Key Rotation - No Password) or Unix Account (Privileged Account SSH Key Rotation - No Password) template on a secret, see "Managing Secrets" on page 1340.



Unix Account (SSH) Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Unix Account (SSH) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

With this Secret Server feature, admins can use private SSH keys for PuTTY launcher sessions, RPC tasks (configurable through password changer settings), and Unix and Linux discovery. Passphrases can additionally be stored, if necessary, to decrypt the private keys for additional security. The Unix Account (SSH) secret template includes text-entry fields for the private key and passphrase by default.

The SSH Key template is included by default and can be used to store SSH keys that can later be selected for RPC, discovery, or launcher authentication for other secrets.

The **Unix Account (SSH)** secret template uses password changers that change the public key in the account's `authorized_keys` file and the account password. Secret Server ships with a password changer and custom command sets that allow an account to change its public key and password, as well as a password changer and custom command sets that change a user's public key and password using a privileged account. These scripts can be customized for different Unix environments.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Unix Account (SSH), we want the Unix Account (SSH) template.

You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a Unix Account (SSH) secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a Unix Account (SSH) secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that a Unix Account Custom (SSH) RPC refers to the Unix Account (SSH) secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Unix Account (SSH) template on a secret, see "Managing Secrets" on page 1340.

Unix Account (Telnet) Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Unix Account (Telnet). With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Secret Server can use scripted password changers for devices that support SSH or Telnet (this allows for flexibility in changing passwords on less common devices). You can also run custom RPC PowerShell scripts to change other platforms' passwords.

With the help of the Secret Server, admins can use private SSH keys for PuTTY launcher sessions and RPC tasks—configurable through password changer settings. Secret Server supports SSH key rotation on secrets.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Unix Account (Telnet), we

want the Unix Account (Telnet) template.

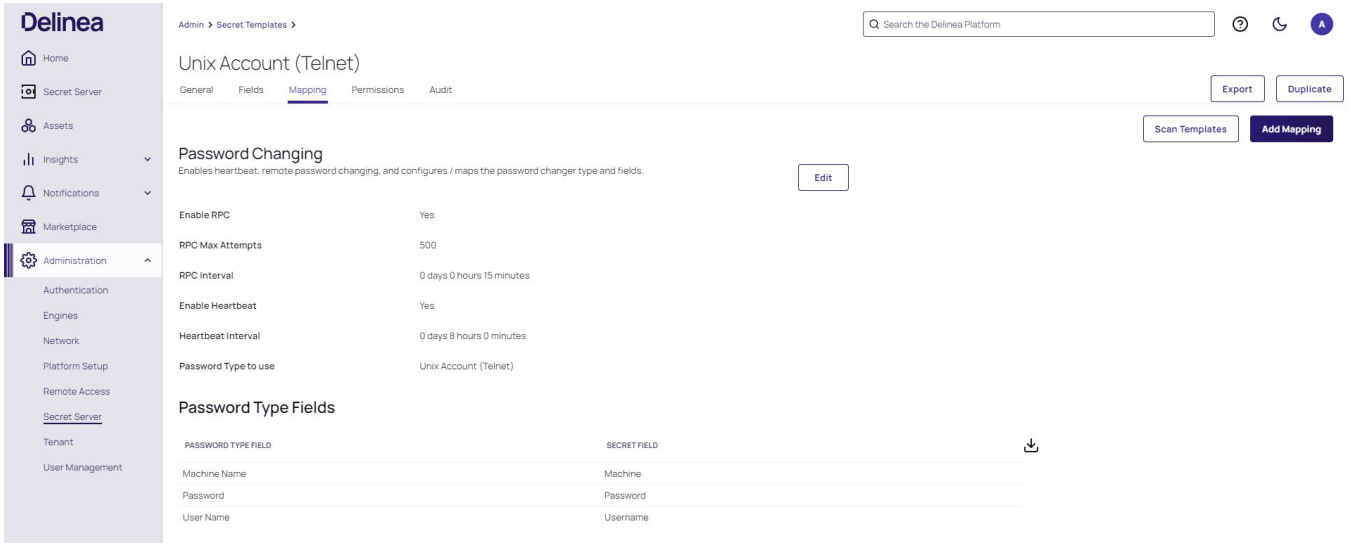
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a Unix Account (Telnet) secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a Unix Account (Telnet) secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that a Unix Account (Telnet) RPC refers to the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To utilize the Unix Account (Telnet) template on a secret, see "Managing Secrets" on page 1340.



Unix Root Account (SSH) Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for the Unix Root Account (SSH). With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

With this Secret Server feature, admins can use private SSH keys for PuTTY launcher sessions, RPC tasks (configurable through password changer settings), and Unix and Linux discovery. Passphrases can additionally be stored, if necessary, to decrypt the private keys for additional security. The Unix Account (SSH) secret template includes text-entry fields for the private key and passphrase by default.

The SSH Key template is included by default and can be used to store SSH keys that can later be selected for RPC, discovery, or launcher authentication for other secrets.

The **Unix Root Account (SSH)** secret template uses password changers that change the public key in the account's `authorized_keys` file and the account password. Secret Server ships with a password changer and custom command sets that allow an account to change its public key and password and a password changer and custom command sets that change a user's public key and password using a privileged account. These scripts can be customized for different Unix environments.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Unix Root Account (SSH), we want the Unix Root Account (SSH) template.

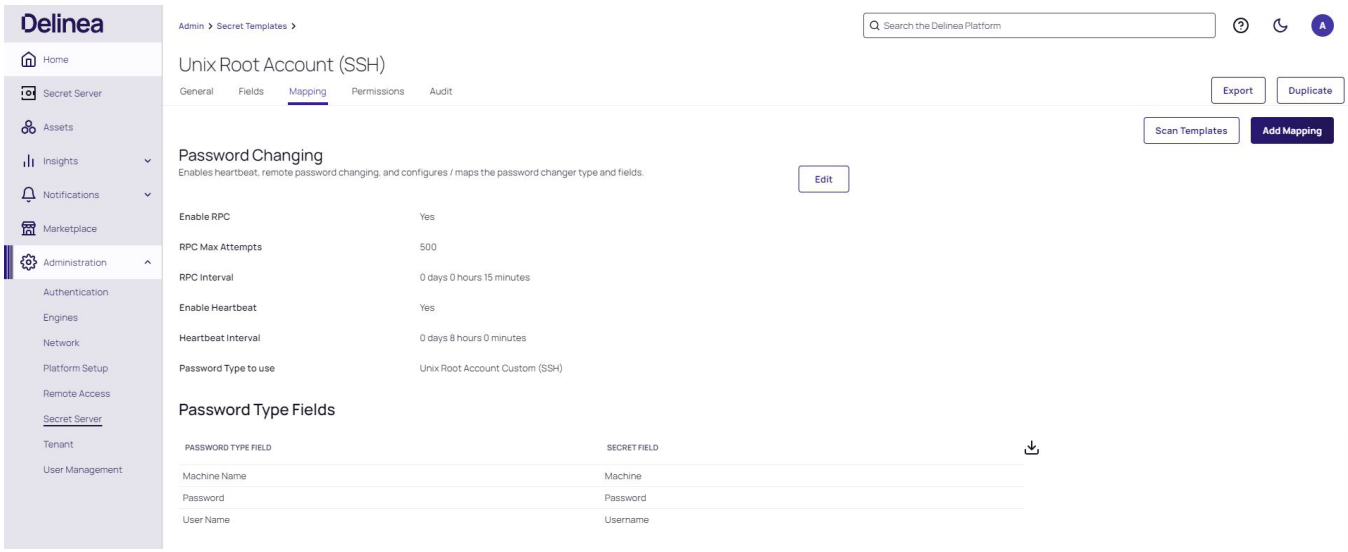
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a Unix Root Account (SSH) secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a Unix Root Account (SSH) secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that a Unix Root Account Custom (SSH) RPC refers to the Unix Root Account (SSH) secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Unix Root Account (SSH) template on a secret, see "Managing Secrets" on page 1340.



VMware Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for the VMware ESX/ESXi (API) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

During configuration, Secret Server is given a list of IP addresses or computer names that correspond to ESX or ESXi servers. Secret Server then connects to each server using the provided credentials to query for a list of user accounts on the target system. For details, see "VMware ESX/ESXi Account Discovery" on page 649.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the VMware ESX/ESXi (API), we want the VMware ESX/ESXi (API) template.

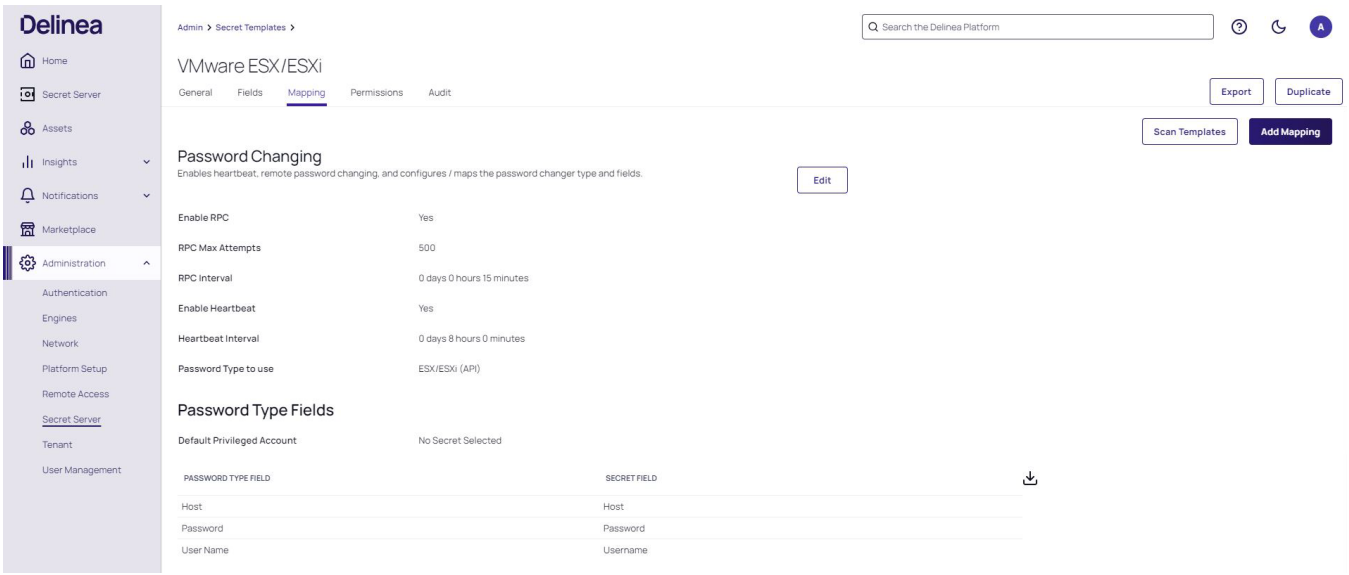
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a VMware ESX/ESXi (API) secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a VMware ESX/ESXi secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that a VMware ESX/ESXi (API) RPC refers to the ESX/ESXi secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the VMware ESX/ESXi template on a secret, see "Managing Secrets" on page 1340.



WatchGuard Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for the WatchGuard accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

WatchGuard is a technology company that specializes in network security products and services, including firewalls, secure Wi-Fi, multi-factor authentication, and network intelligence solutions for small to medium-sized businesses and organizations.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the WatchGuard, we want the WatchGuard template.

You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

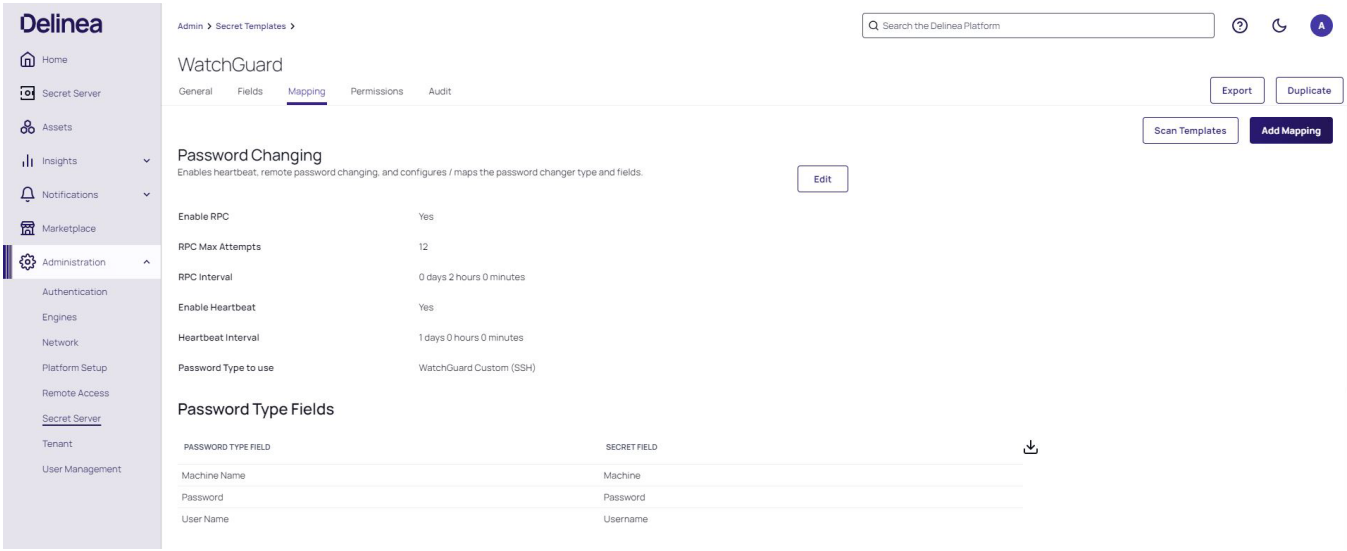
To navigate to a WatchGuard secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a WatchGuard secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that a

WatchGuard Custom (SSH) RPC refers to the WatchGuard secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the WatchGuard template on a secret, see "Managing Secrets" on page 1340.



Web Password Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for the Web Password accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Web Password, we want the Web Password template.

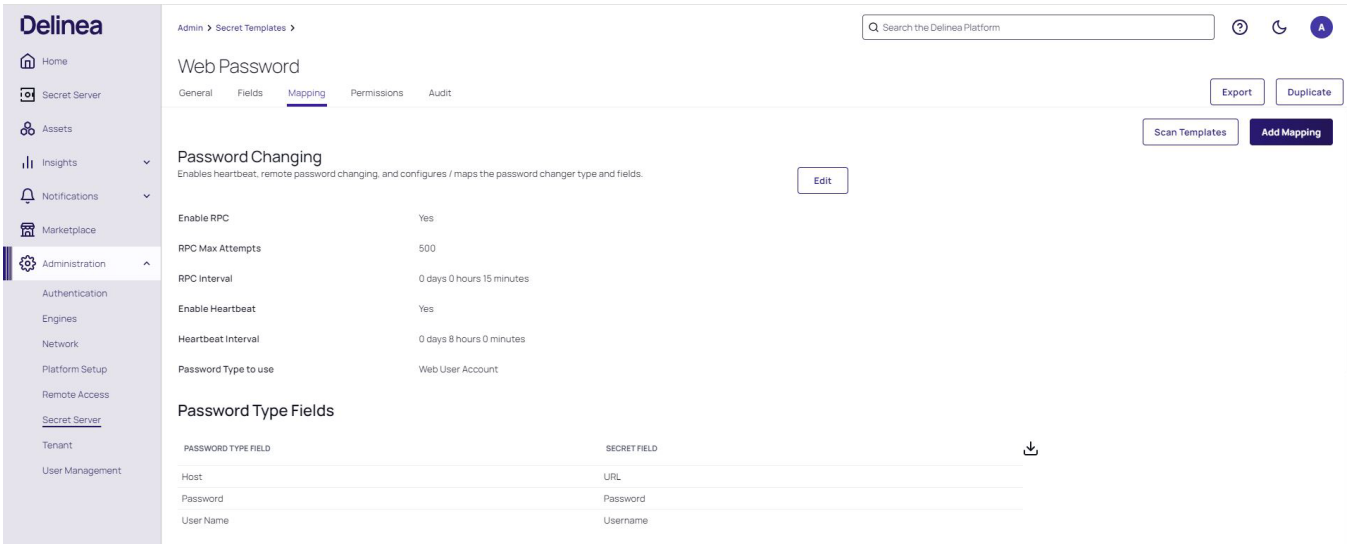
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a Web Password secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a Web Password secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that a Web User Account RPC refers to the Web Password secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Web Password template on a secret, see "Managing Secrets" on page 1340.



Windows Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for the Windows Account. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Windows Account, we want the Windows Account template.

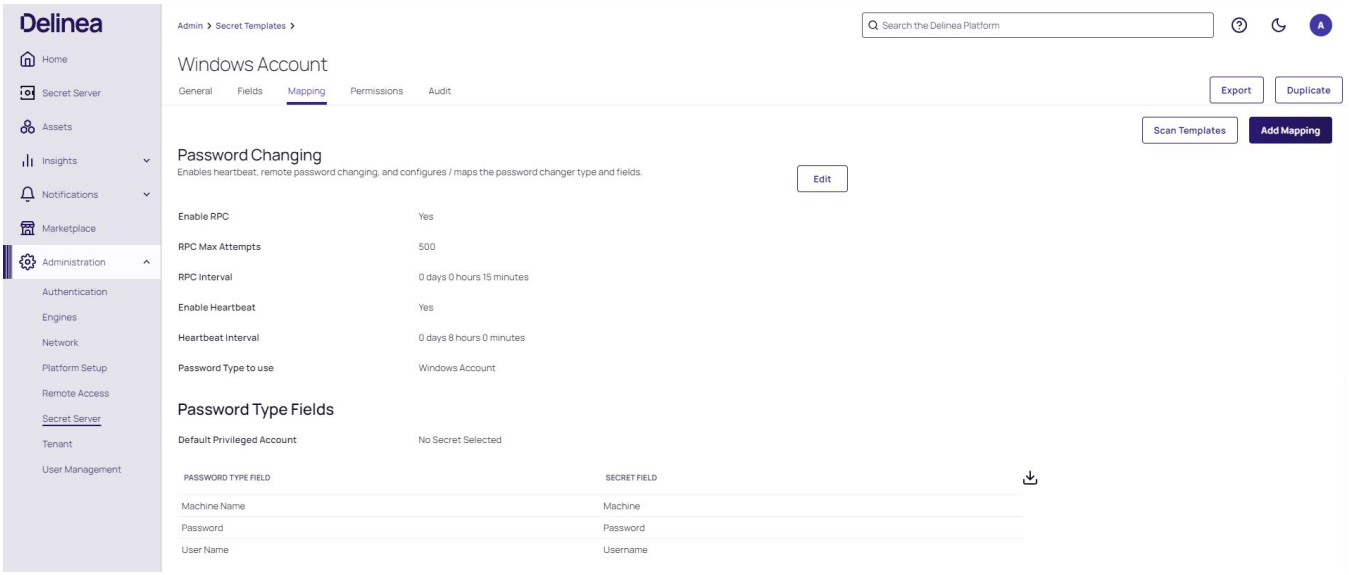
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to a Windows Account secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select a Windows Account secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that a Windows Account RPC refers to the Windows Account secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Windows template on a secret, see "Managing Secrets" on page 1340.



RPC for Specific Vendors and Technologies

Remote Password Changing (RPC) in Secret Server supports a wide range of specific vendors and technologies, enabling automated password management across diverse systems. Each vendor or technology, such as ServiceNow, WatchGuard, HP iLO, Sybase, Okta, IBM iSeries (AS/400), VMware ESX/ESXi, and SonicWall NSA, has a dedicated secret template preconfigured with the appropriate password changer. These templates ensure that passwords can be automatically updated when they expire or on a defined schedule, maintaining synchronization with Secret Server. Administrators can also customize these templates and password changers to fit specific requirements, leveraging SSH, Telnet, or PowerShell scripts for more complex environments. This flexibility allows organizations to securely manage credentials across various platforms, enhancing overall security and operational efficiency.

See the Delinea Integrations site for more about vendors and technologies.

This section of topics is about configuring RPC for specific technologies, not about using templates to create secrets. For that, see "Included RPC Templates" on page 951.

RPC for Active Directory

Active Directory Remote Password Changing (RPC) in Secret Server allows for the automated management of AD account passwords, ensuring they remain synchronized with Secret Server policies. This feature supports automatic password updates upon expiration or on a defined schedule, adhering to domain password policy

requirements. To enable AD RPC, administrators must configure the appropriate permissions for the service account, ensuring it has the necessary rights to change passwords. Secret Server provides a dedicated Active Directory Account template preconfigured for this purpose, which can be customized as needed. Additionally, the platform supports advanced configurations such as using privileged accounts for password changes and integrating PowerShell scripts for more complex environments. This robust functionality ensures secure and efficient password management across Active Directory environments.

Setting Minimum Permissions for AD RPC Service Accounts

Overview

Secret Server requires proper permissions to perform remote password changing (RPC). The privileged Delinea Secret Server RPC service account used for RPC of an Active Directory (AD) account secret, must have granular permissions applied to it. You will be using two Active Directory tools to make these modifications to the RPC account:

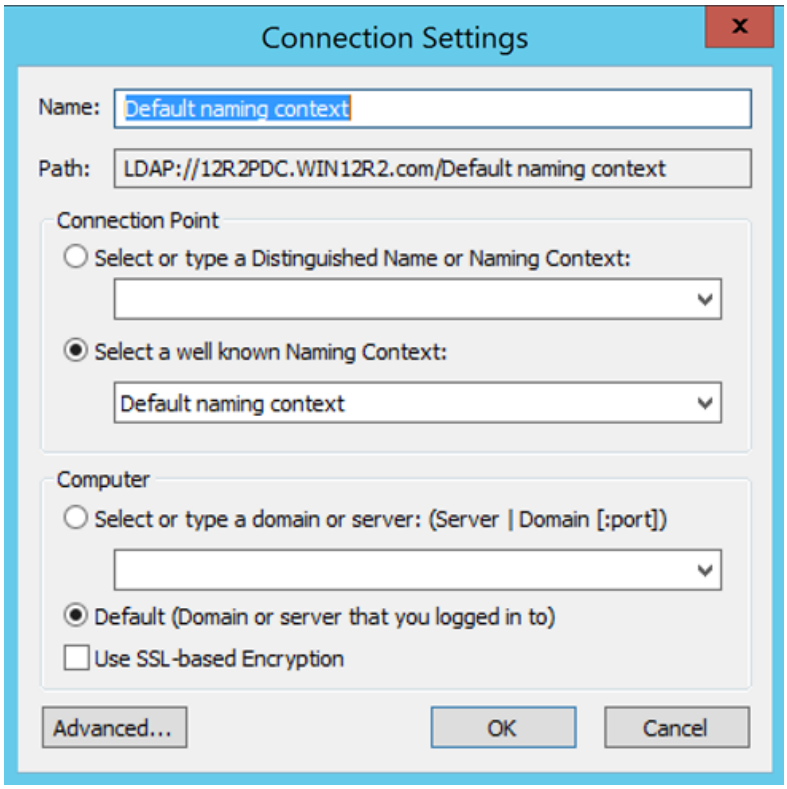
- ADSI Edit
- Active Directory Users and Computers.

> ADSI stands for *Active Directory Service Interfaces*. It is a set of COM interfaces used to access the features of directory services from different network providers.

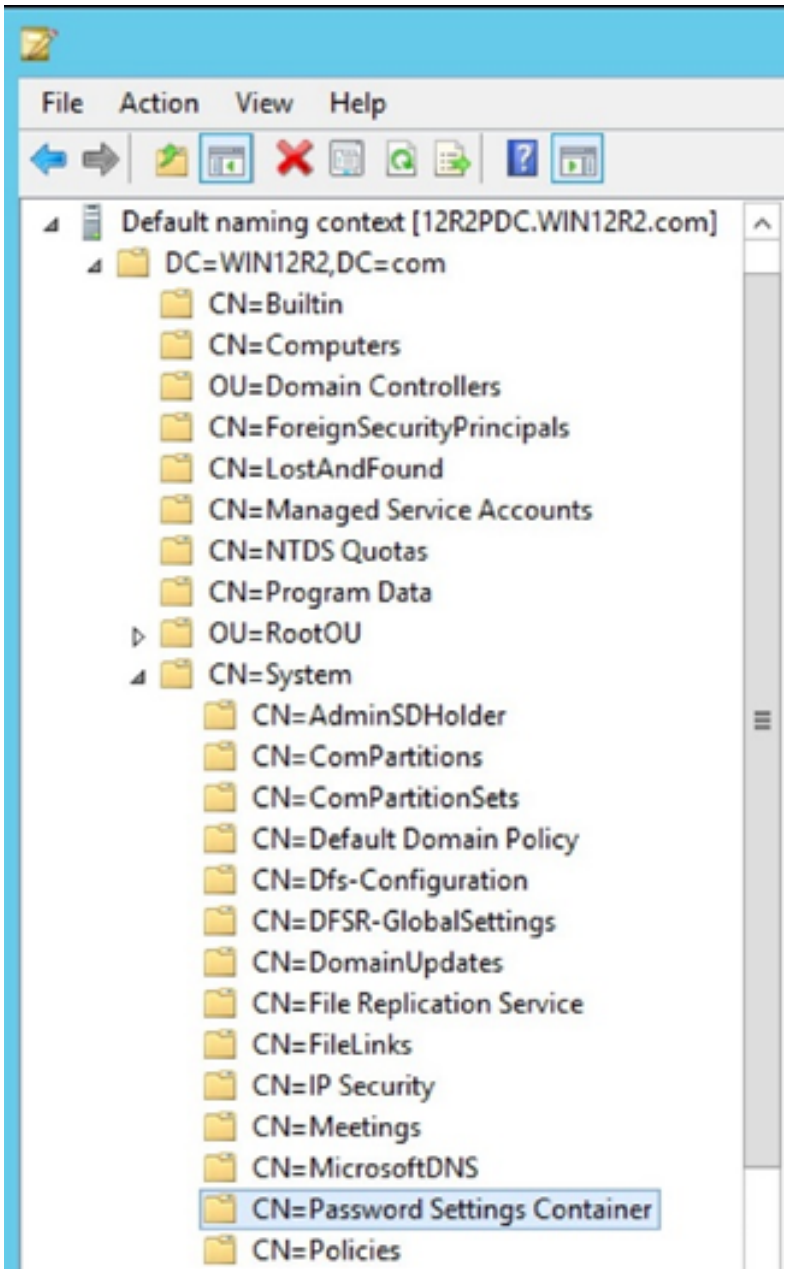
> The AD password changer has an RPC **timeout minutes** advanced setting. This setting only applies when using **Password Change By Admin Credentials**.

Setting ADSI Permissions

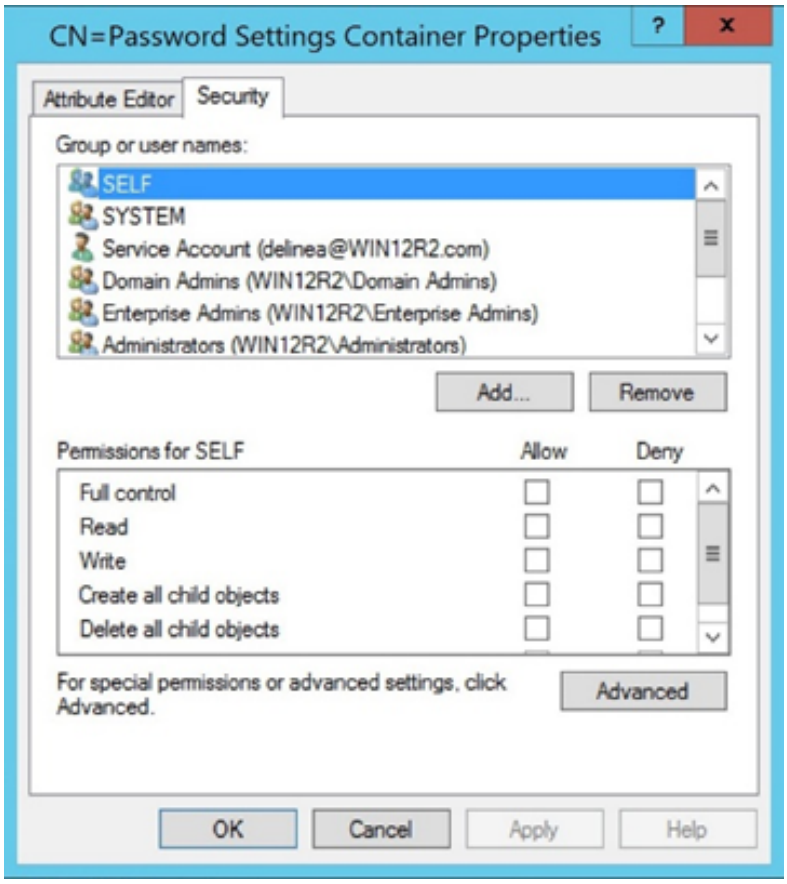
1. Open ADSI Edit (found on Domain Controllers as part of the Active Directory Administration Tools).
2. From the **Action** drop down menu select **Connect to...**. The "Connection Settings" window appears:



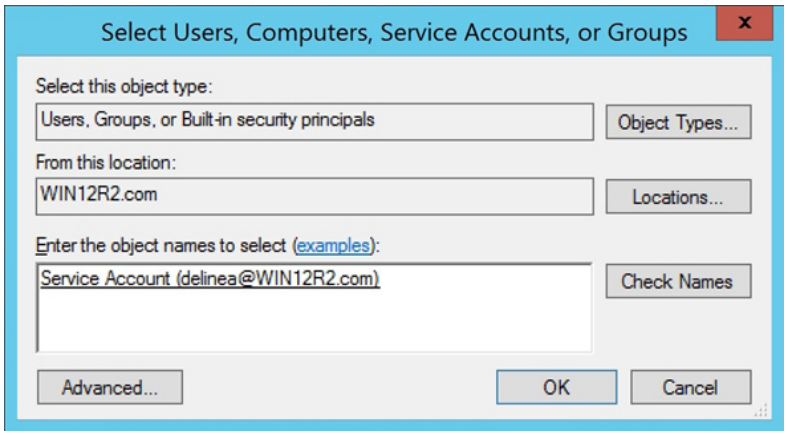
3. Make any adjustments if needed.
4. Click the **OK** button to connect to the domain you are logged into. The ADSI Edit window appears.
5. Click on the **Default naming context** node (the root of the domain).
6. Expand the domain name root and scroll maneuver down until you reach **CN=System > CN=Password Settings Container** as noted in the image below:



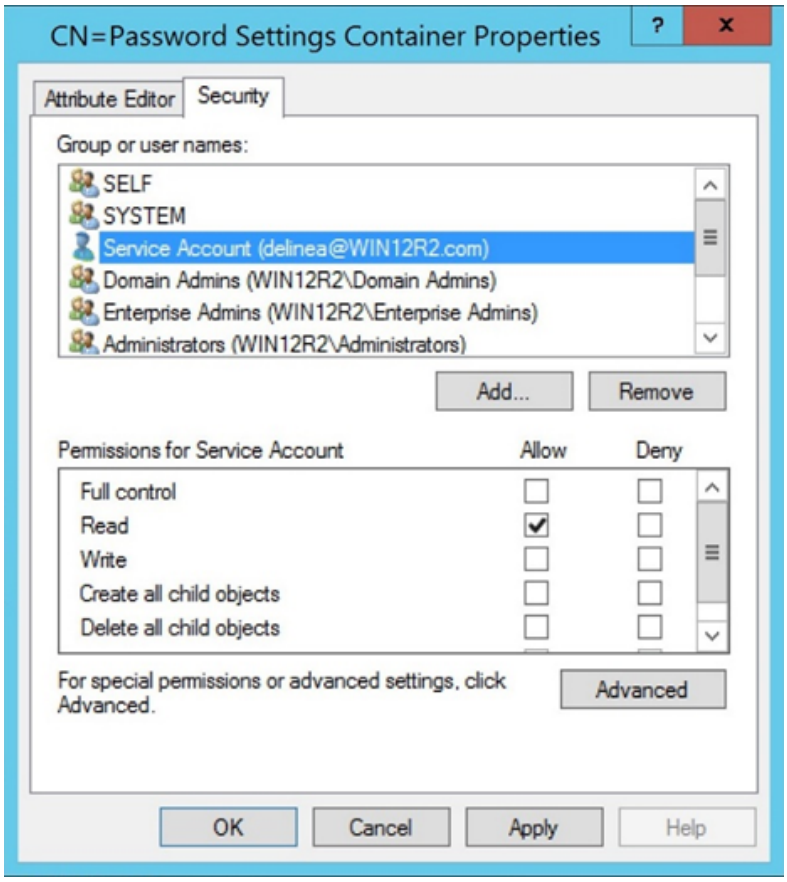
7. Right-click **CN=Password Settings Container** and select **Properties**. A properties dialog box appears:



8. Click the **Add...** button. The "Select Users, Computers, Service Accounts, or Groups" dialog box appears:



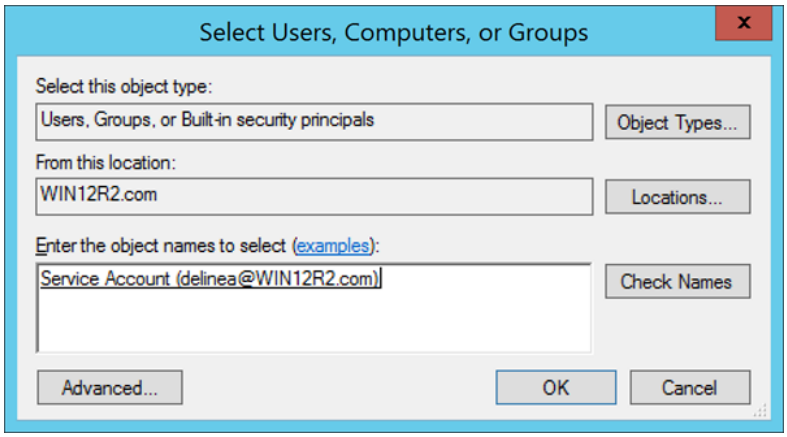
9. Type the information for the Delinea Secret Server RPC account.
10. Click the **OK** button. The previous dialog box reappears with the "delinea" service account appearing in the "Group or user names" list.
11. Click on the new account, its permissions appear:



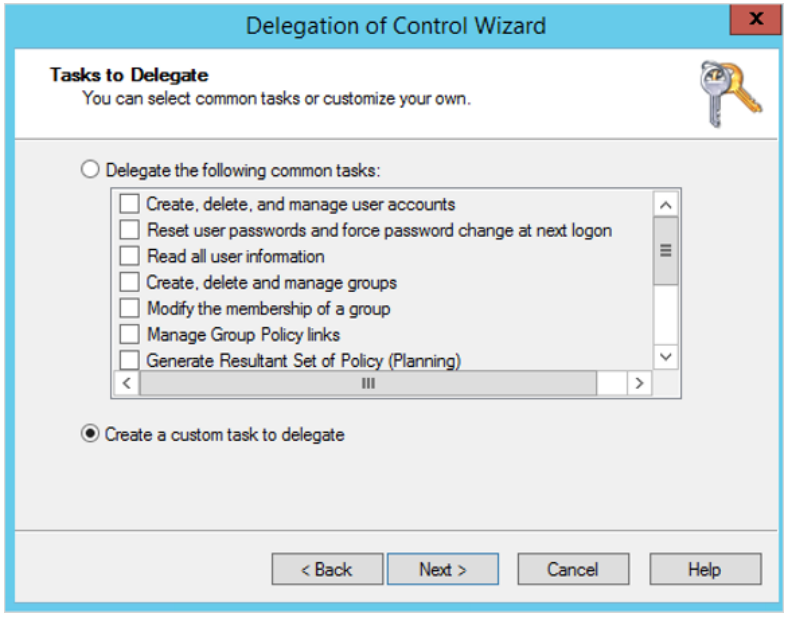
12. Click to select the **Read** check box in the **Allow** column.
13. Click the **OK** button.

Setting Delegate Control Permissions

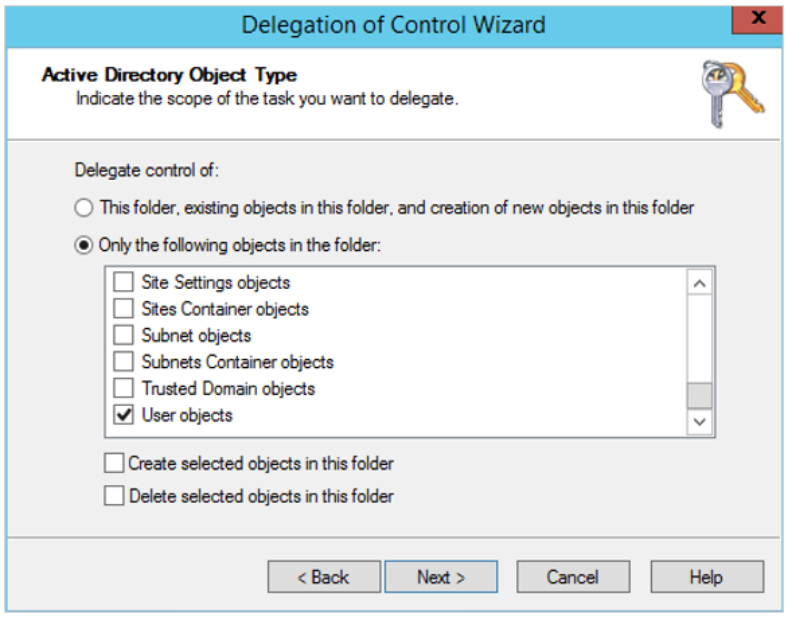
1. Open the **Active Directory Users and Computers** administrative console.
2. Right-click the Organizational Unit (OU) or the top-level domain you want to configure and select **Delegate Control...** as noted in the image below. The "Delegation of Control Wizard" appears.
3. Click the **Next** button. The **Users or Groups** dialog appears.
4. Click the **Add...** button in the **Users or Groups** section.
5. Click the **Add...** button. The "Select Users, Computers, Service Accounts, or Groups" dialog box appears:



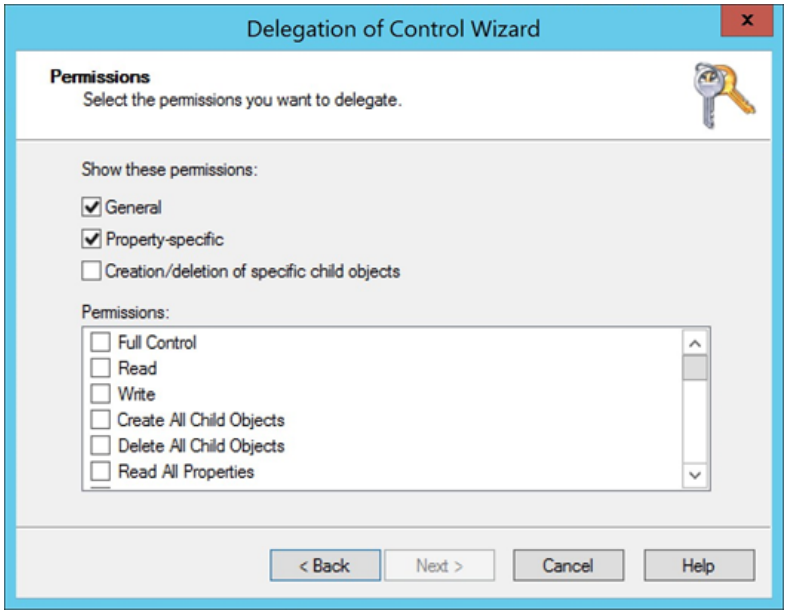
6. Type the information for the Delinea Secret Server RPC account.
7. Click the **OK** button. The Wizard reappears.
8. Click the **Next** button. The Tasks to Delegate page of the wizard appears:



9. Click to select the **Create a custom task to delegate** selection button.
10. Click the **Next** button. The Active Directory Object Type page of the wizard appears:



11. Click to select the **Only the following objects in the folder** selection button.
12. Scroll to bottom of the list.
13. Click to select the **User objects** check box.
14. Click the **Next** button. The Permissions page of the wizard appears:



15. Click to select the **General** and **Property-specifics** check boxes.
16. In the **Permissions** list, ensure none of the check boxes are selected.
17. Locate and click to select the following check boxes in the **Permissions** list:

- Change Password
- Reset Password
- Read lockoutTime
- Write lockoutTime
- Read pwdLastSet
- Write pwdLastSet
- Read UserAccountControl
- Write UserAccountControl

18. Click the **Next** button.
19. Click the **Finish** button.

Setting Delegate Control Permissions for Protected Group Accounts

See the Protected Accounts and Groups in Active Directory page for details about protected groups.

Configure delegation control in the domain controller for protected group accounts:

1. At a command prompt on the domain controller, type the following command to grant the domain account permission to unlock the account:

`dc=cps` and `dc=com` in the following commands should be changed to your domain name.

```
dsacls "dc=cps,dc=com" /G "<yourDomainName>\<yourAccountName>:RP;msDS-User-Account-
Control-Computed;user" /I:S
dsacls "dc=cps,dc=com" /G "<yourDomainName>\yourAccountName>:RPWP;lockoutTime;user" /I:S
dsacls "CN=AdminSDHolder, CN=System, DC=cps, DC=com"  /G
"<yourDomainName>\<yourAccountName>:RPWP;lockoutTime"
```

2. At a command prompt on the domain controller, type the following command to grant the domain account permission to reset the password:

`dc=cps` and `dc=com` in the following commands should be changed to your domain name.

```
dsacls "dc=cps,dc=com" /G "<yourDomainName>\<yourACcountName>:CA;Reset Password;user"
/I:S
dsacls "CN=AdminSDHolder, CN=System, DC=cps, DC=com" /G
"<yourDomainName>\<yourAccountName>:CA;Reset Password"
```

It can take a while for the Security Descriptor Propagator Update (SDProp) process to pick up the new settings from AdminSDHolder.

To initiate the SDProp process immediately, complete the following steps:

1. Click **Run** and type `ldp.exe` in the domain controller desktop Start menu.
2. Select **Connection > Connect...** from the LDP window.
3. In the Connect window, make sure **389** is listed in the **Port** field, and then click **OK**.
4. Select **Connection > Bind...** from the LDP window.
5. Select **Bind as currently logged on user** and click **OK**.
6. Select **Browse > Modify** from the LDP window.
7. Configure the following fields in the Modify window:
   - **DN**: empty
   - **Attribute**: `RunProtectAdminGroupsTask`
   - **Values**: 1
   - **Operation**: click **Add** and then click **Enter**.
8. Click **Run**.

If you have a large environment, it may take some time for SDProp to update the protected admin group permissions.

Configuring Delegation Control for Administrative Accounts

If you use a regular domain account (not part of the Domain Admins group) for the administrative account, you need to configure the domain account delegation in the domain controller.

> The delegated permissions configured for the administrative account are not available for some protected groups. See Delegated permissions are not available and inheritance is automatically disabled, for details.

> To enable delegated permissions on the administrative account in order to manage protected groups, see the additional configuration steps in "Configuring Delegation Control for Administrative Accounts" above.

Configuring Delegation Control for the Administrative Account

To configure delegation control in the domain controller for the administrative account, do the following:

1. In the domain controller of the domain, select **Administrative Tools > Active Directory Users and Computers**.
2. Right-click the domain with the accounts that will be managed. Select **Delegate Control**, then click **Next** at the Welcome window.
3. In **Users and Groups**, click **Add** and enter the name of the user you want to configure with the administrative account (with unlock and password reset permissions) and click **OK**.
4. In **Task to Delegate**, select **Create a custom task to delegate** and click **Next**.

5. In **Active Directory Object Type**, select **Only the following objects in the folder**, as well as **User objects**, then click **Next**.
6. In **Permissions**, select the following:
   a. **General** and **Reset password** to delegate password reset rights.
   b. **Property-specific, Read msDS-User-Account-Control-Computed, Read lockout Time**, and **Write lockout Time** to delegate account unlock rights.
7. Click **Next** and then **Finish**.

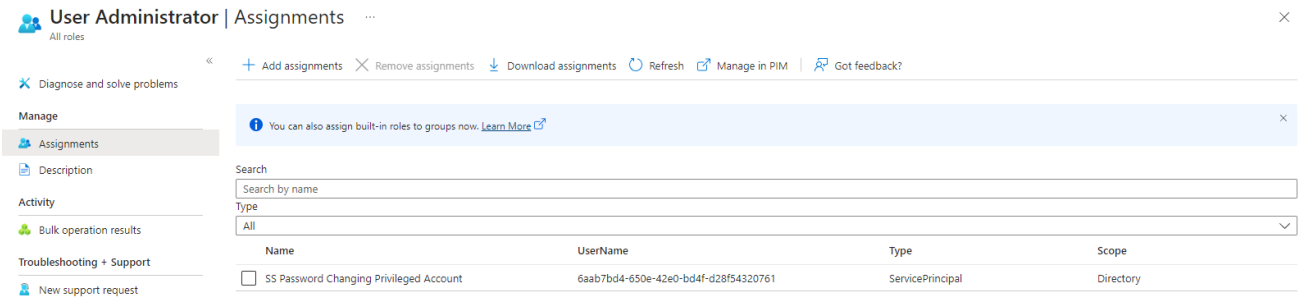
The domain account with delegated permissions can now be configured as the domain administrative account for the account unlock and automatic account maintenance features.

Minimum Permissions for Entra ID RPC

> For the complete setup for Entra ID RPC, see "Configuring an Azure AD or Entra ID Password Changer" on the next page.

Secret Server requires proper permissions to perform remote password changing (RPC). The privileged Delinea Secret Server RPC service principal used for RPC of an Entra ID user account secret, must be assigned to the User Administrator role.

1. Log into the Entra ID or Azure AD Portal (https://portal.azure.com).
2. Go to **Microsoft Entra ID > Roles and Administrators**.
3. Select the **User Administrator** role.
4. Click **Add Assignments**.



5. Search for the desired service principal. This is the account to give permissions to, in this case, the registered application.
6. Click **Add**.

> Please note, that these permissions will only work for non-administrator accounts. For administrator accounts, users need to have at least *Privileged Authentication Administrator* permissions. For more information about the Entra ID secret template, see "Entra ID Secret Template for RPC" on page 958.

Configuring an Azure AD or Entra ID Password Changer

The built-in templates use Microsoft Graph in Entra ID (Azure AD). The password changer works based on a combination of items including:

- Entra ID tenant ID.
- Client ID (originates from an app registration in Azure AD or Entra ID).
- Client secret (originates from an app registration in Azure AD or Entra ID).
- Username of the managed user in UPN format (username@domain).
- Password of the managed user.
- New password of the managed user generated by the Secret Server password changer.

The built-in templates are designed to perform the password change as the application registration itself, in order to facilitate managing MFA-protected accounts. The heartbeat functionality is designed for both MFA and non-MFA protected accounts using error-handling logic.

> The authentication flow is based on the resource owner password credentials grant flow. This flow is not recommended for regular use but is being used in this case because the script is designed to run as a service account, and the user account being interacted with is managed by secret-server. For password changing and validation, the actual account credentials are acquired without human interaction. This enables the script to manage accounts that are also protected with MFA.

Requirements

> See "Minimum Permissions for Entra ID RPC" on the previous page for information on the required permissions.

- A valid Entra ID tenant.
- A dedicated app registration for the password changer:
  - See Register an application with the Microsoft identity platform for more information on creating an app registration. See "Task 1: Creating an App Registration" below for details.
  - Azure Application Registration template to store application registration information in secret-server. This comes with the out-of-the-box integration.
  - Entra ID User Account template to store Entra ID secrets. This also comes with the out-of-the-box integration.
- The username should be stored in UPN format (username@domain.com) for the password changer to function.
- User Administrator permissions are needed to configure Entra ID RPC.

Task 1: Creating an App Registration

The application registration provides the password changer with the permissions to perform the password change. The following steps are required to create the application registration:

1. Log into the Entra ID or Azure AD Portal (https://portal.azure.com).
2. Navigate to **Microsoft Entra ID**.
3. Navigate to **App Registrations**.
4. Click **New Registration**.
5. Provide a name for the application registration. You will need this name later in the instruction.
6. Select the account type (single tenant or multi-tenant).
7. Optionally, provide a redirect URI.
8. Click **Register**. Make note of the application (client) ID, which is required when creating the corresponding secret in secret-server.
9. While still on the **App Registration** page, navigate to **Manage > Certificates & Secrets**.
10. Click **New Client Secret**.
11. Provide a description for the client secret.
12. Select an expiration date.
13. Click **Add**.
    Make note of the client secret value, which is required when creating the corresponding secret in **Secret Server**.
14. Navigate to **API Permissions**.
15. Check if the following API permission is assigned: **User.Read**. If not assigned add the permission by:
    a. Clicking **Add a Permission**.
    b. Selecting **Microsoft Graph**.
    c. Selecting **Delegated Permissions**.
    d. Choosing the **User.Read** permission.
16. Click on **Grant Admin Consent** for the respective tenant button. This completes the creation of the application registration.

Task 2: Providing the Password Changer App the Permissions to Manage Users

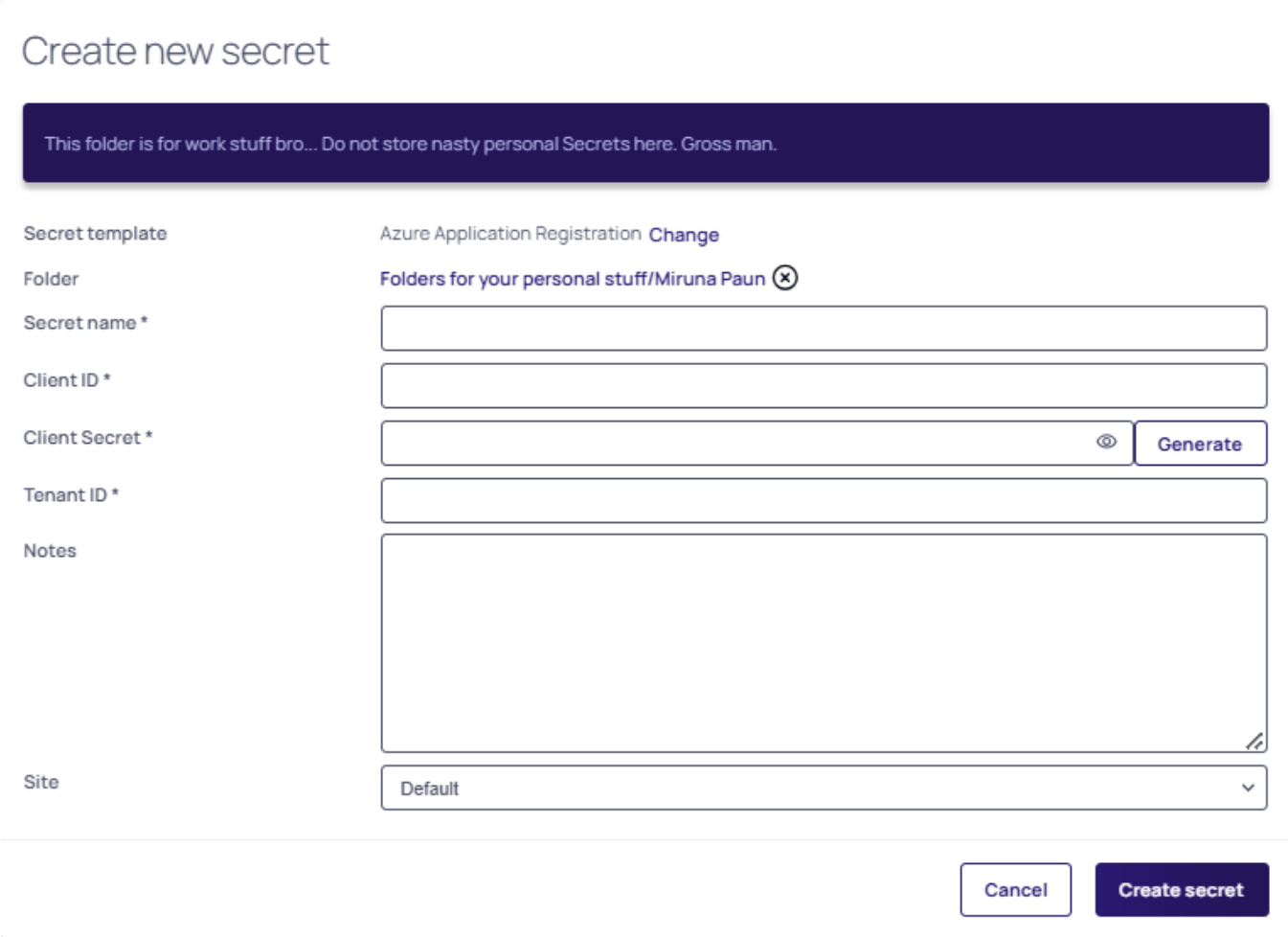
See "Minimum Permissions for Entra ID RPC" on page 1001 for more on this topic.

1. If necessary, log into the Azure Portal (https://portal.azure.com)
2. Navigate to **Microsoft Entra ID**.
3. Navigate to **Manage > Roles and Administrators**.
4. In the search bar, search for the role (such as Global Administrator). We recommend the **Privileged Authentication Administrator** role.
5. Click the role name in the resulting list.
6. Click **Add Assignments**.

7. Search for and select the previously created application registration.
8. Click **Add**.

Task 3: Creating an Azure App Registration Secret in secret-server

1. Log into secret-server or platform.
2. Navigate to **Secrets**.
3. Click **Create Secret**.
4. Select the template to store the application registration information (**Azure Application Registration**).
5. Fill out the required fields with the information from the application registration:
   - Secret Name (for example, Delinea Entra ID Password Changer).
   - Client ID.
   - Client Secret, which you created earlier when creating the app registration (recorded in an task earlier step).
   - Tenant ID, which you can retrieve from Entra ID, specifically, the Directory (tenant) ID property.
   - Application ID, can be added to the notes for reference. You can retrieve this ID from the **Application Registration > Application (client) ID** (noted down in an task earlier step).
6. Click **Create Secret**.



Task 4: Associating the Privileged Account with the Entra ID Account Secret

To correctly use the password changer, the privileged account must be associated with the Entra ID User Account secret:

1. Log into secret-server or platform.
2. Navigate to **Secrets**.
3. Locate your secret(s) based on the Entra ID User Account template.
4. Click on the secret.
5. Click **Remote Password Changing**.
6. Select **Privileged Account Credentials** on the Change Password Using selection button.
7. Click **No Secret Selected**. A list of eligible secrets appears.
8. Search for and select the earlier-created **Azure Application Registration** secret.
9. Click **Save**.

Task 5: Testing the Configuration

If all went well, you now should have:

- A secret in secret-server for the application registration.
- An Entra ID User Account secret (not covered in this guide).
- The application registration secret associated with the Entra ID User Account secret.

To test the configuration, you can start with performing a heartbeat on the Entra ID User Account secret:

1. Log into secret-server.
2. Navigate to **Secrets**.
3. Locate your secret based on the Azure AD Account template.
4. Click the secret.
5. Click **Heartbeat**. After a few moments the heartbeat should complete successfully.

To further test the configuration, you can change the password of the Entra ID User Account secret:

1. Log into secret-server.
2. Navigate to **Secrets**.
3. Locate your secret based on the Entra ID User Account template.
4. Click the secret.
5. Click **Change Password Now**.
6. Select **Randomly Generated** or **Manual** (and enter a password).
7. Click **Change Password**.

Creating a Custom Password Changer for Cisco ASA

To create a custom password changer using SSH for Cisco ASA 5505, 5515 and other models with IOS 12.2 and earlier that cannot use the copy command, follow the procedure for "Creating a Custom Password Changer" on page 905. Make sure you choose a base password changer that ends with (SSH) with a command set similar to the one you are adding, and use the following settings:

Authenticate As

1. `$[1]$USERNAME`
2. `$[1]$PASSWORD`

Commands

1. Enter `enable`
2. Enter `$CURRENTPASSWORD`
3. Enter `config terminal`
4. Enter `enable password $NEWPASSWORD`

5. Enter `end`
6. Enter `wr mem`
7. Enter `exit`

Creating a Custom Password Changer for IBM AS/400

> Password changing on the IBM AS/400 can be performed through SSH, which is installed by default. If you are using an earlier version, you will need to install SSH.

To create a custom password changer for IBM AS/400 on newer systems such as i7, use the procedure for "Creating a Custom Password Changer" on page 905 but be sure to use the following SSH command:

- Command: system `CHGUSRPRF $USERNAME PASSWORD($NEWPASSWORD)`
- Comment: Set Password on account
- Pause(ms): 2000

> For additional information, see Securing Communications with OpenSSH on IBM i5/OS.

Creating a Custom Password Changer for IBM AS/400 in Secret Server 10.5.

The procedure for creating password changers in Secret Server 10.5 for the IBM AS/400 terminal includes using the 5250 terminal connection and scripting to perform the password change and heartbeat.

To create this IBM AS/400 password changer, start with an existing z/OS Mainframe password changer as a baseline, then modify the changer commands. You also need to create an AS/400 secret template using the z/OS secret template as a baseline, then modify the template to use the new password changer.

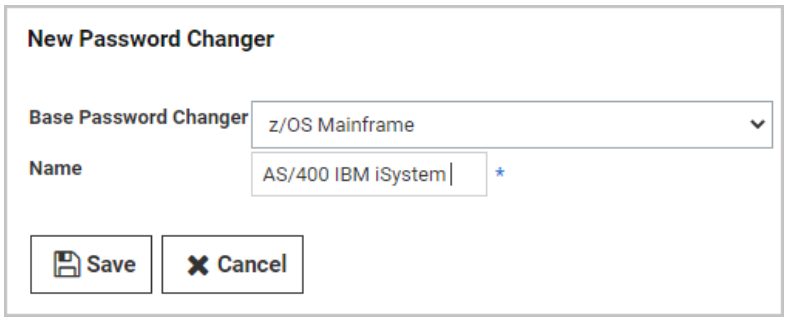
> Support for this feature, including script customization for advanced requirements, is available only through professional services.

Configuration

Follow the procedure below, in the sequence presented.

Create an AS/400 password changer from an existing z/OS Mainframe password changer:

1. Browse to **Admin > Remote Password Changing > Configure Password Changers**.
2. Scroll to the bottom and select **New**.
3. For the **Base Password Changer**, select the **z/OS Mainframe**.
4. In the **Name** field, enter `AS/400 IBM iSystem`

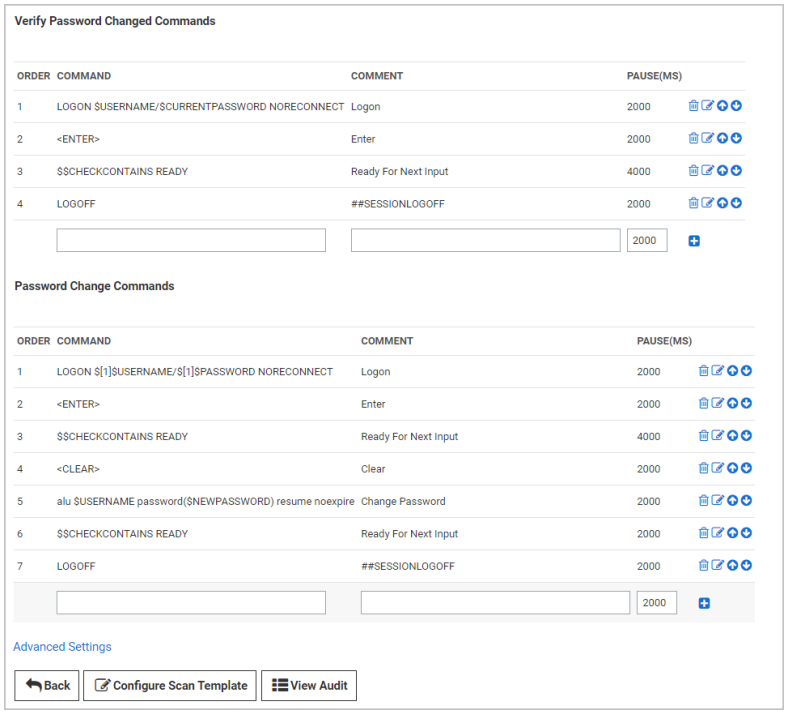


5. Click **Save**.

Modify the AS/400 IBM iSystem password changer commands:

To add custom password changer commands to the AS/400, you must replace the existing standard z/OS mainframe command set.

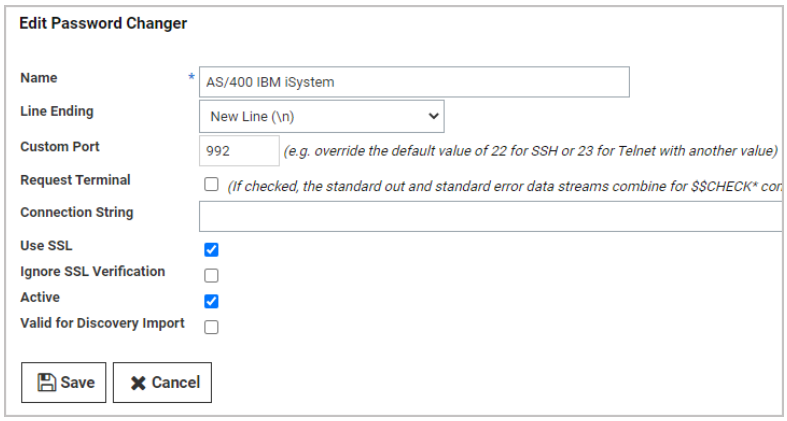
1. Browse to **Admin > Remote Password Changing > Configure Password Changers**.
2. Click the **AS/400 IBM iSystem** password changer you just created.
3. On the **AS/400 IBM iSystem** page, scroll to the bottom and click the **Edit Commands** button. The commands that appear initially on the **Verify Password Changed Commands** page represent the standard z/OS Mainframe command set. You can use these commands as a baseline but you must customize them to suit your environment.



4. Click the **Back** button when you have finished customizing your password changer commands, to return to the **AS/400 IBM iSystem** password changer page.

Modify the AS/400 password changer for 5250 emulation and commands:

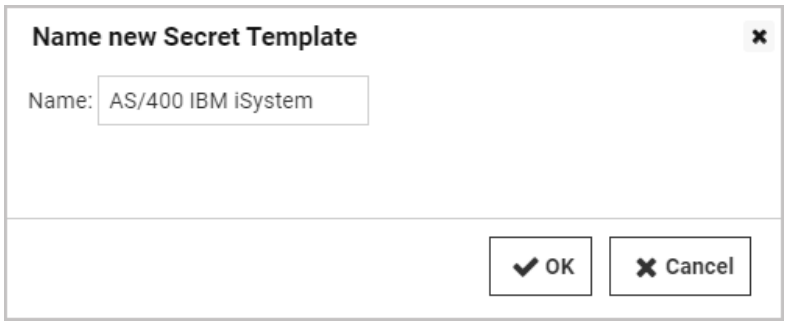
1. On the **AS/400 IBM iSystem** page, scroll to the bottom and click the **Edit** button.
2. On the **Edit Password Changer** page, check the box next to **Use SSL** (recommended).
3. Set the **Custom Port** to `992`.



> For extra troubleshooting assistance, you can add TRACE to the connection string to have a trace file written to the Secret Server website or engine.

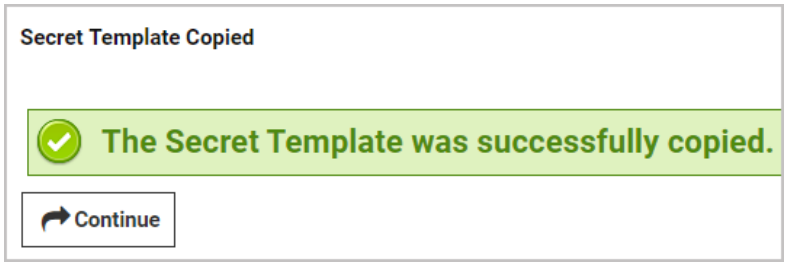
Create an AS/400 template from the z/OS Secret Template:

1. Browse to **Admin > Secret Templates**.
2. On the **Manage Secret Templates** page, select **z/OS Mainframe** from the drop-down menu and click **Edit**.
3. On the **Secret Template Designer** page, scroll to the bottom and click **Copy Secret Template**.
4. On the **Name New Secret Template** page, enter `AS/400 IBM iSystem` in the **Name** field.



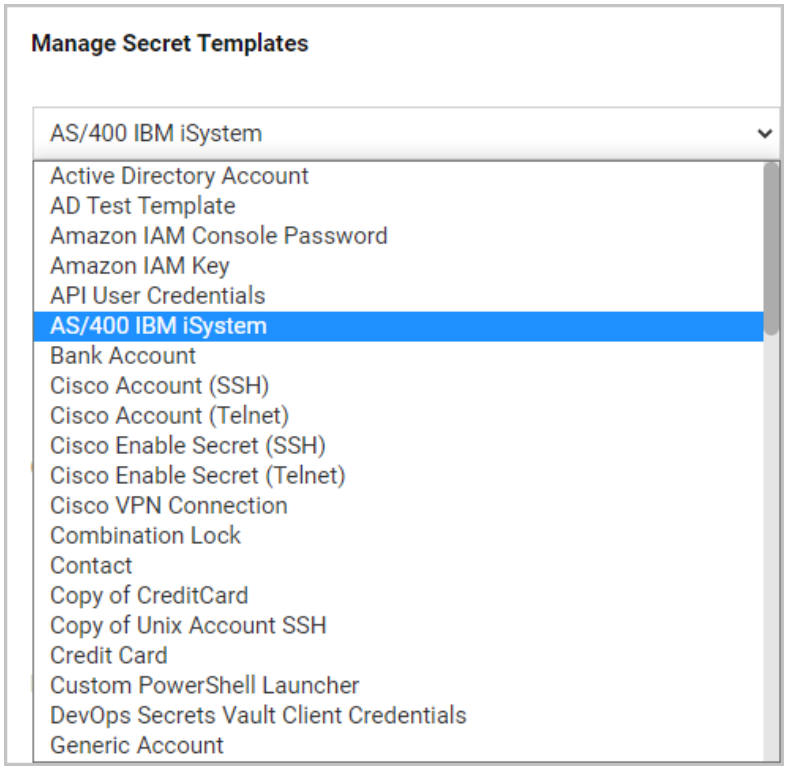
5. Click **OK**.

6. On the confirmation screen, click **Continue**.

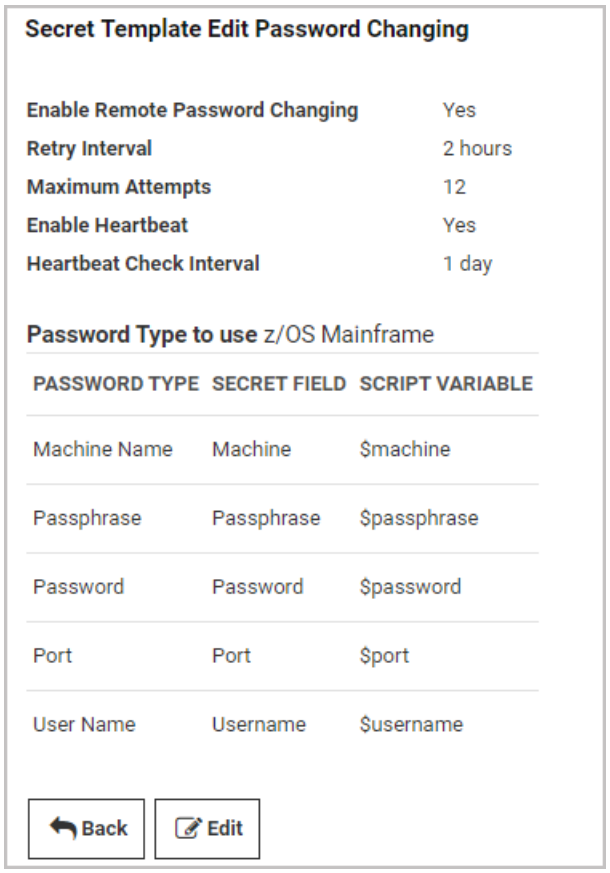


Modify the AS/400 Secret Template to use the AS/400 Password Changer:

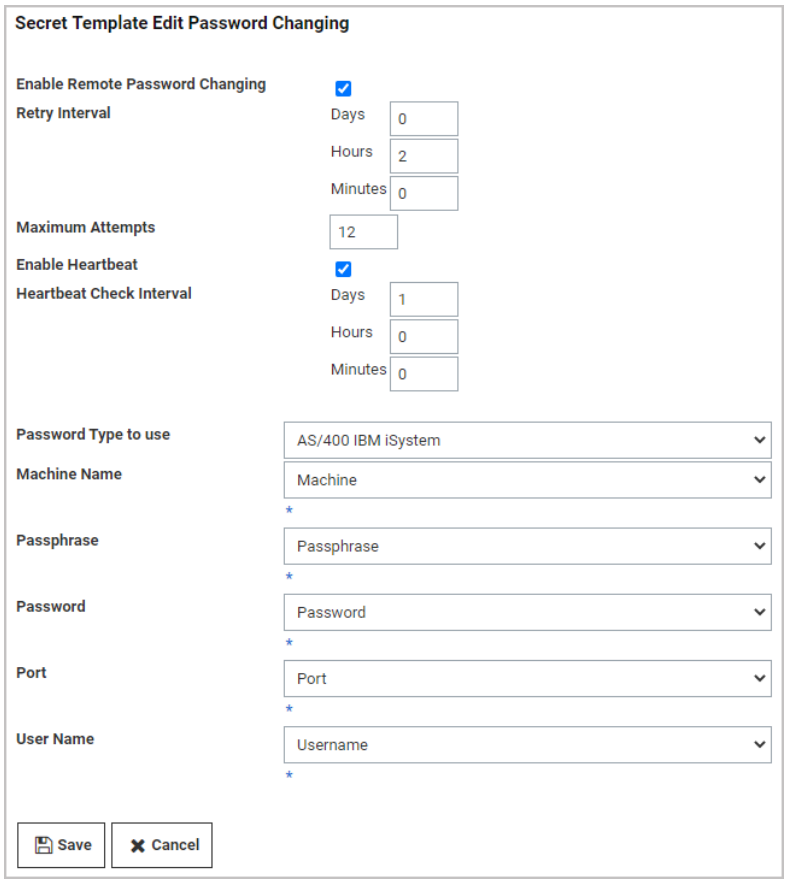
1. Browse to **Admin > Secret Templates**.
2. On the **Manage Secret Templates** page, click the drop-down menu and select the new **AS/400 IBM iSystem** secret template you just created.



3. Click **Edit**.
4. On the **Secret Template Designer** page, scroll to the bottom and click **Configure Password Changing**.
5. On the **Secret Template Edit Password Changing** page, click **Edit**.



6. On the second **Secret Template Edit Password Changing** page, select **AS/400 IBM iSystem** from the **Password Type to Use** drop-down menu.



7. Click **Save**.

Your can now create secrets using your new template and password changer in Secret Server 10.5.

If you would like to test your template and password changer but you do not have access to an AS/400 IBM iSystem, you can use the website PUB400.com.

Running Heartbeat and RPC for Office 365 and Azure Accounts with PowerShell

This page is deprecated. Please refer to additional documentation from Microsoft for more information.

This feature does not require the advanced scripting add-on license. To perform heartbeat checks and remote password changes on secrets using the Office 365 and Azure password changer for user accounts, follow the steps below:

Procedure

This applies to both the Secret Server Web server or distributed engines.

1. Run Windows PowerShell as an admin. This opens an elevated Windows PowerShell command prompt.
2. Run this command: `Install-Module -Name AzureAD`.

3. Recycle the application pool.

> These steps are required once on the subject computer, not every time you connect. However, you may want to update the module periodically as a security best practice using the command: `Update-Module -Name AzureAD`

Troubleshooting

1. Uninstall AzureAD using command `remove-module AzureAD`.
2. Reinstall using the above procedure.
3. Ensure the Secret Server application pool setting *Load User Profile* is set to "True".
4. Recycle the application pool.

Oracle RPC Templates

Secret Server now has four Oracle templates, three current and one legacy:

- Oracle Account
- Oracle Account (TCPS)
- Oracle Account (Template Ver 2)
- Oracle Account (Walletless)

See "Configuring Oracle Secret Templates" on page 1205 for details.

Configuring Oracle DB 19c for Heartbeat and RPC

This document explains how to configure Oracle Database 19c for heartbeat and remote password changing (RPC) with Secret Server. It consists of installing the Oracle Database Access Components (ODAC), configuring Secret Server, and configuring one or more distributed engines.

> This document is not updated with every release–many releases do not affect the guide's contents and thus do not warrant a document update.

This Delinea technical configuration knowledge base article is relevant to and has been tested on:

- Secret Server 10.7 on Windows Server 2016 Standard (64-bit)
- Distributed engine 10.7 on Windows Server 2016 Standard (64-bit)
- Oracle Database 19c on Windows Server 2019 Standard (64-bit)

Introduction

This document explains how to configure Oracle Database 19c for heartbeat and remote password changing (RPC) with Secret Server. The process consists of installing the Oracle Database Access Components (ODAC) and configuring Secret Server and one or more distributed engines.

Procedure

Task One: Installing the Oracle Database Access Components

1. Navigate to ODAC Runtime Downloads in your browser.

   Note: Oracle Account (Template Ver 2) does not actually require this extension to be installed

2. Download the latest version of the ODAC OUI file with the same major number as your database version.
3. Unzip the file.
4. Right click and run setup.exe as a Windows administrator. The setup wizard appears.
5. Click to select **Use Windows Built-in Account**.
6. Click the **Next** button.
7. Type the desired installation path, such as `c:\oracle`.
8. Click the **Next** button.
9. Confirm the default product components are selected. If not, click to select **Reset Defaults**.
10. Click the **Next** button.
11. Leave the **DB Connection Configuration** fields as they are (empty).
12. Click the **Next** button. The setup runs some pre-installation configuration tests.
13. When the tests are completed, click the **Install** button.
14. Reboot your machine.

Task Two: Configuring Secret Server

1. Navigate to the ODAC directory, such as c:\oracle.
2. Copy the `C:\oracle\product\19.x.x\client_1\odp.net\bin\4\Oracle.DataAccess.dll` file to the bin directory of your Secret Server directory, for instance, `c:\inetpub\wwwroot\SecretServer\bin`.
3. Navigate to the `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config` directory.
4. Open the `machine.config` file.
5. Copy and paste the line below into the `DbProviderFactories` section:

```
<add name="Oracle.DataAccess" invariant="Oracle.DataAccess.Client" description="Oracle Data Provider for .NET, Unmanaged Driver" type="Oracle.DataAccess.Client.OracleClientFactory, Oracle.DataAccess, Version=4.122.19.1, Culture=neutral, PublicKeyToken=89b483f429c47342" />
```

The section should look like this:

```
<system.data>   <DbProviderFactories>      <add name="Oracle.DataAccess" invariant="Oracle.DataAccess.Client" description="Oracle Data Provider for .NET, Unmanaged Driver" type="Oracle.DataAccess.Client.OracleClientFactory, Oracle.DataAccess, Version=4.122.19.1, Culture=neutral, PublicKeyToken=89b483f429c47342"/>
</DbProviderFactories></system.data>
```

There may be additional <Add> sections, such as for Microsoft SQL Server. Leave them as is.

Task Three: Configuring a Secret Server Distributed Engine

1. Install ODAC on the machine hosting the distributed engine using the same procedure as Task One.
2. Navigate to the ODAC directory on the distributed engine machine, such as `c:\oracle`.
3. Copy the `C:\<ODAC_Directory>\odp.net\bin\4\Oracle.DataAccess.dll` file to the Distributed Engine directory, for instance, `C:\Program Files\Thycotic Software Ltd\Distributed Engine`.
4. Navigate to the `C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Config` directory.
5. Open the `machine.config` file.
6. If necessary, create a `<DbProviderFactories>` section within the <system.data> section.
7. Copy and paste the line below into the `<DbProviderFactories>` section:

```
<add name="Oracle.DataAccess" invariant="Oracle.DataAccess.Client" description="Oracle Data Provider for .NET, Unmanaged Driver" type="Oracle.DataAccess.Client.OracleClientFactory, Oracle.DataAccess, Version=4.122.19.1, Culture=neutral, PublicKeyToken=89b483f429c47342"/>
```

The section should look like this:

```
<system.data>   <DbProviderFactories>      <add name="Oracle.DataAccess" invariant="Oracle.DataAccess.Client" description="Oracle Data Provider for .NET, Unmanaged Driver" type="Oracle.DataAccess.Client.OracleClientFactory, Oracle.DataAccess, Version=4.122.19.1, Culture=neutral, PublicKeyToken=89b483f429c47342"/>
</DbProviderFactories></system.data>
```

Troubleshooting

Log Files

The errors below may appear in these files:

- `SS-BWSR.log` (Secret Server)
- `SSDE.log` (distributed engine)

These files are located within the log subdirectory of the application's directory. Typically, these are:

- Secret Server: `c:\inetpub\wwwroot\secretserver\log\SS-BWSR.log`
- Distributed engines: `c:\program files\thycotic software ltd\distributed engine\log\SSDE.log`

Errors

Oracle.DataAccess.Client.OracleException: The provider is not compatible with the version of Oracle client

This error occurs when the Oracle ODAC driver does not match the Oracle database version.

Uninstall the ODAC, and then re-install the correct version. You can uninstall ODAC using the universal installer that is included with the ODAC installation that resides in the ODAC directory.

Oracle.DataAccess.Client.OracleException (0x80004005): ORA-12514: TNS:listener does not currently know of service requested in connect descriptor

This error occurs when the secret's database field does not match the Oracle SERVICE_NAME database.

> The default "Oracle Account" secret template's database field is looking for the Oracle SERVICE_NAME database. You can find that database's location by reading the tnsnames.ora configuration file on your Oracle database server.

System.ArgumentException: Unable to find the requested .Net Framework Data Provider. It may not be installed.

This error occurs when Oracle parameters are missing from the section in the machine.config file.

System.Configuration.ConfigurationErrorsException: Unrecognized element

This error occurs when the section, located in the machine.config file, is not properly formatted.

Configuring Oracle Secret Templates

Secret Server now has four Oracle templates, three current and one legacy:

- Oracle Account
- Oracle Account (TCPS)
- Oracle Account (Template Ver 2)
- Oracle Account (Walletless)

Introduction

Overview

Secret Server now has three secret templates based off the Oracle Managed Data Access NuGet library. Unlike earlier Oracle secret templates, templates, using the NuGet library does not require `Oracle.ManagedDataAccess.dll` to be installed alongside Secret Server or its engines. Additionally, two of these templates support Oracle's TCPS connection protocol. You can run the new templates alongside the earlier Oracle secret templates, but we recommend using the new templates when creating new Oracle secrets.

DataSource Field

All three new templates include an optional DataSource field. The DataSource field acts like a connection string to the Oracle database. When used, it is not necessary to fill out the Host, Database, Port, or SSL Server Cert DN fields. On the secret template's page, "none" appears in each of those fields.

Without DataSource:

Secret Name *	Oracle Walletless without DataSource
Secret Template	Oracle Account (Walletless)
Host	adb.us-ashburn-1.oraclecloud.com
Username *	walletless_local
Password *	**********
Database	g62a2e091eede11_jjtest2_high.adb.oraclecloud.com
Port	1521
As System User? *	0
DataSource	None
SSL Server Cert DN	CN=adwc.uscom-east-1.oraclecloud.com, OU=Oracle BMCS US, O=Oracle Corporation, L=Redwood City, ST=California, C=US
Notes	None

With DataSource:

Secret Name *	Walletless with DataSource
Secret Template	Oracle Account (Walletless)
Host	None
Username *	walletless_with_datasource
Password *	**********
Database	None
Port	None
As System User? *	0
DataSource	(description= (retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1521)(host=adb.us-ashburn-1.oraclecloud.com))(connect_data=(service_name=g62a2e091eede11_jjtest2_high.adb.oraclecloud.com))(security=(ssl_server_dn_match=yes)(ssl_server_cert_dn="CN=adwc.uscom-east-1.oraclecloud.com, OU=Oracle BMCS US, O=Oracle Corporation, L=Redwood City, ST=California, C=US")))
SSL Server Cert DN	None
Notes	None

As System User Field

All new templates also have the “As System User” field. It allows connections to the database as the system user. Setting this to “1” allows database connections with the SYSDBA privilege. Setting this to “0” connects with the default role.

Templates

Oracle Account

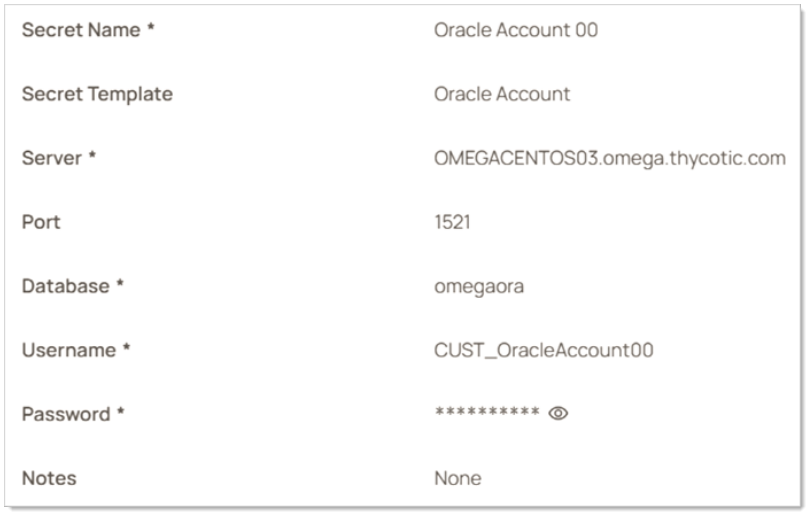
This is the deprecated original template that maintained for legacy implementations. I requires that `Oracle.ManagedDataAccess.dll` is installed alongside Secret Server or its engines.

Oracle Account (Template Ver 2)

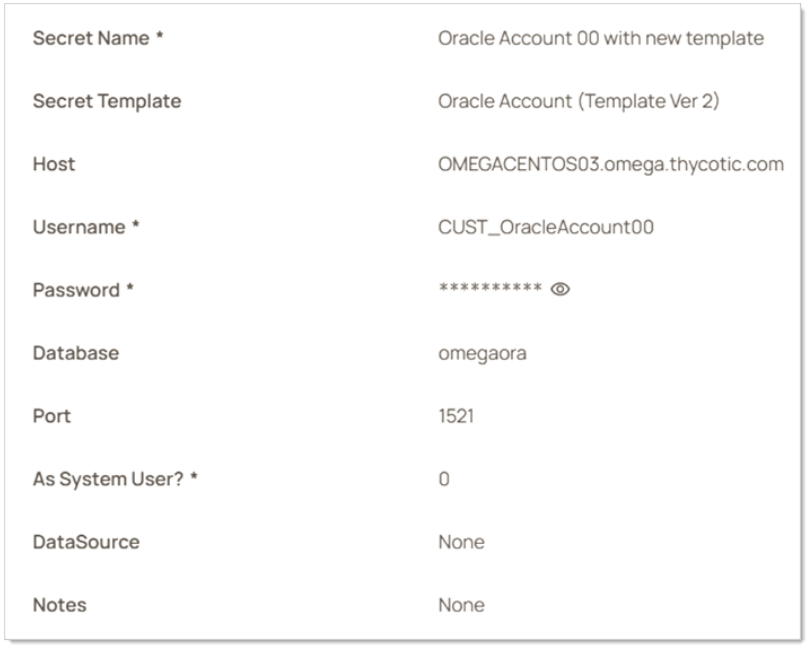
Oracle Account (Template Ver 2) is the closest equivalent to the original Oracle Account template. It does not support Oracle TCPS, but it does use the Oracle Managed Data Access NuGet library. The fields on this secret

template are the same as in the original Oracle Account template with a few exceptions. We added two new fields, the DataSource and “As System User” fields described above. Additionally, the “Server” field is called “Host” in the new template to more closely match the terminology in Oracle’s connection string.

Original template:



New template:



See "Oracle Account Secret Template for RPC" on page 1021 for more using this template.

Oracle Account (TCPS)

Overview

You can make TCPS connections using the "Oracle Account (TCPS)" secret template and an Oracle Wallet. As described by Oracle, “Oracle Wallet is a container that stores authentication and signing credentials.”

> See Understanding Oracle Wallet for more information, and refer to the documentation on your specific Oracle database for detains on obtaining and using your wallet.

Wallet Location

Prior to setting up Oracle Account (TCPS) secrets, you will need to place copies of your Wallet files on the same server(s) as your Secret Server site(s) where you will have Oracle Account (TCPS) secrets. Afterwards, use the “Wallet Location” field to note the location of your wallet files on your newly created secret. Note that you will need to ensure that the user running Secret Server’s app pool (or engine when applicable) is granted permissions to access the directory where the wallet is located.

TNS Admin

The “TNS Admin” field is optional. If you have a `tnsnames.ora` file, specify its containing directory in this field.

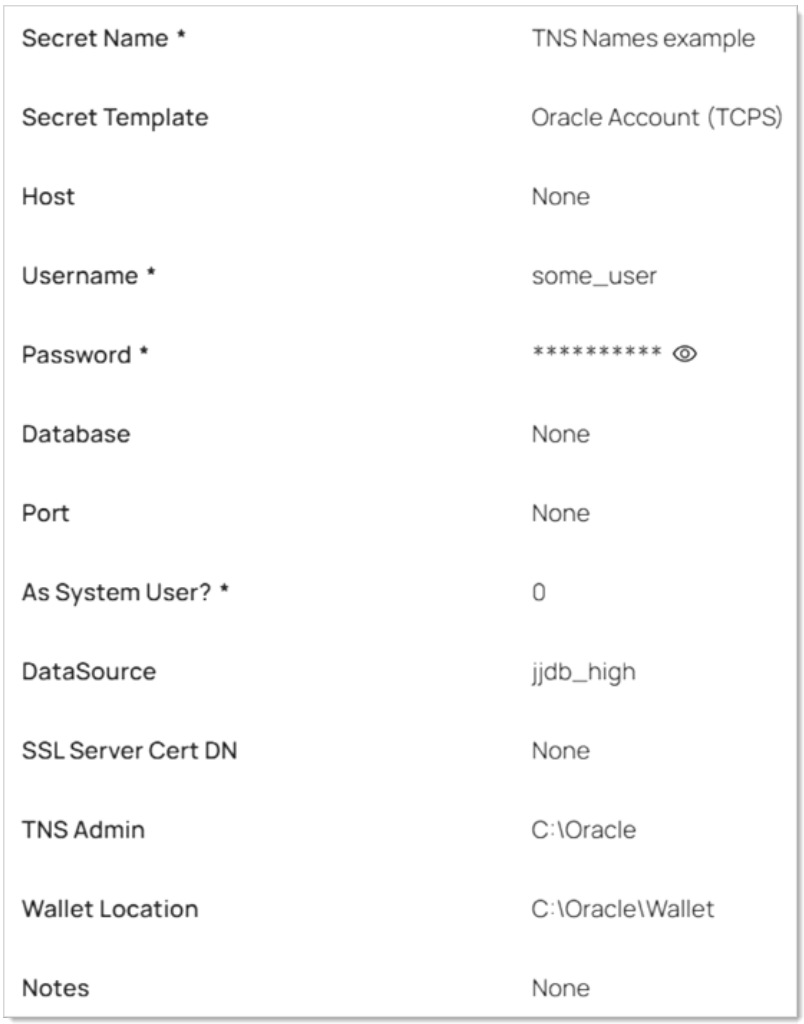
As with wallets, you should consult documentation specific to your Oracle database for information on using `tnsnames.ora`. However, in general, `tnsnames.ora` file is a configuration file containing network service names mapped to connect descriptors (for local naming method) or net service names mapped to listener protocol addresses.

> See Local Naming Parameters in the tnsnames.ora File for more information.

Thus, you can use the `tnsnames.ora` file as an alternative way to specify a connection string. First, put the directory of the `tnsnames.ora` file in the “TNS Admin” field. Second, format the contents of `tnsnames.ora` as `<ALIAS> = <CONNECTION STRING>`. Paste the desired alias from inside the `tnsnames.ora` files in the DataSource field of the secret. For example, if we had a file at `C:\Oracle\tnsnames.ora`, the contents might be as follows:

```
jjdb_high = (description= (retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1522)
(host=adb.us-ashburn-1.oraclecloud.com))(connect_data=(service_name=g62a2e091eede11_jjdb_
high.adb.oraclecloud.com))(security=(ssl_server_cert_dn="CN=adwc.uscom-east-
1.oraclecloud.com, OU=Oracle BMCS US, O=Oracle Corporation, L=Redwood City, ST=California,
C=US")))

jjdb_low = (description= (retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1522)
(host=adb.us-ashburn-1.oraclecloud.com))(connect_data=(service_name=g62a2e091eede11_jjdb_
low.adb.oraclecloud.com))(security=(ssl_server_cert_dn="CN=adwc.uscom-east-1.oraclecloud.com,
OU=Oracle BMCS US, O=Oracle Corporation, L=Redwood City, ST=California, C=US")))
```

In this example, `jjdb_high` or `jjdb_low` are both aliases. We would choose one, say `jjdb_high`, and paste it into the DataSource field on my secret. I would include `C:\Oracle` in the TNS Admin field. My secret would appear as in the following screenshot:



Oracle Account (Walletless)

Oracle has recently announced its support of TCPS without requiring the use of wallets. See Securely Connecting to Autonomous DB Without a Wallet (Using TLS) for details.

Because walletless connections utilize TLS instead of “mutual TLS,” this is potentially less secure than a wallet-based authentication, so you should do your own research before deciding if this is the right approach for you. The advantage of the walletless template is that it allows a secure TCPS connection with the simplicity of the “Oracle Account (Template Ver 2)” template. No wallet files need to be deployed to a server.

The fields on this template are the same as Oracle Account (Template Ver 2) with the addition of the “SSL Server Cert DN” field, which can be found in the TLS connection string. Refer to documentation on your specific Oracle database about how to enable walletless connections, as well as how to obtain the TLS connection string to fill in fields on this secret template.

Oracle Account Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Oracle Account and Oracle Account (Template Ver 2) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

To configure secret templates for Oracle, see "Configuring Oracle Secret Templates" on page 1205.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Oracle, we want the Oracle Account template.

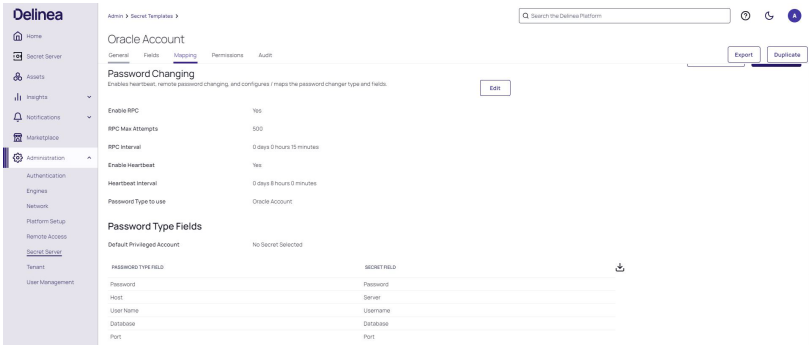
You can view and modify secret Templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to an Oracle Account secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select an Oracle Account secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that the Oracle Account RPC conforms to the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Oracle Account template on a secret, see "Managing Secrets" on page 1340.



Oracle Account (TCPS) Secret Template for RPC

Overview

This document briefly discusses using Secret Server Remote Password Changing (RPC) for Oracle Account (TCPS) and Oracle Account (Walletless) accounts. With Remote Password Changing (RPC), secrets can automatically change remote account passwords when a secret expires, either immediately or on a defined schedule. In addition, the new passwords' strengths and other qualities are completely configurable. See the "Password Changer List" on page 909 for a complete list of available password changers.

Oracle TCPS (Transparent Network Substrate over SSL/TLS) is a technology used in Oracle Database environments to provide secure, encrypted communication between clients and servers. It operates as an extension of Oracle's Transparent Network Substrate (TNS), a fundamental part of the Oracle Net Services, facilitating network sessions from clients to Oracle databases.

To configure secret templates for Oracle, see "Configuring Oracle Secret Templates" on page 1205.

Assigning a Password Changer to a Secret Template

After completing the RPC setup, you can manage the built-in secret templates. Each secret template is specific application and is preconfigured with the password changer best suited to that. For the Oracle Account (TCPS), we want the Oracle Account (TCPS) template.

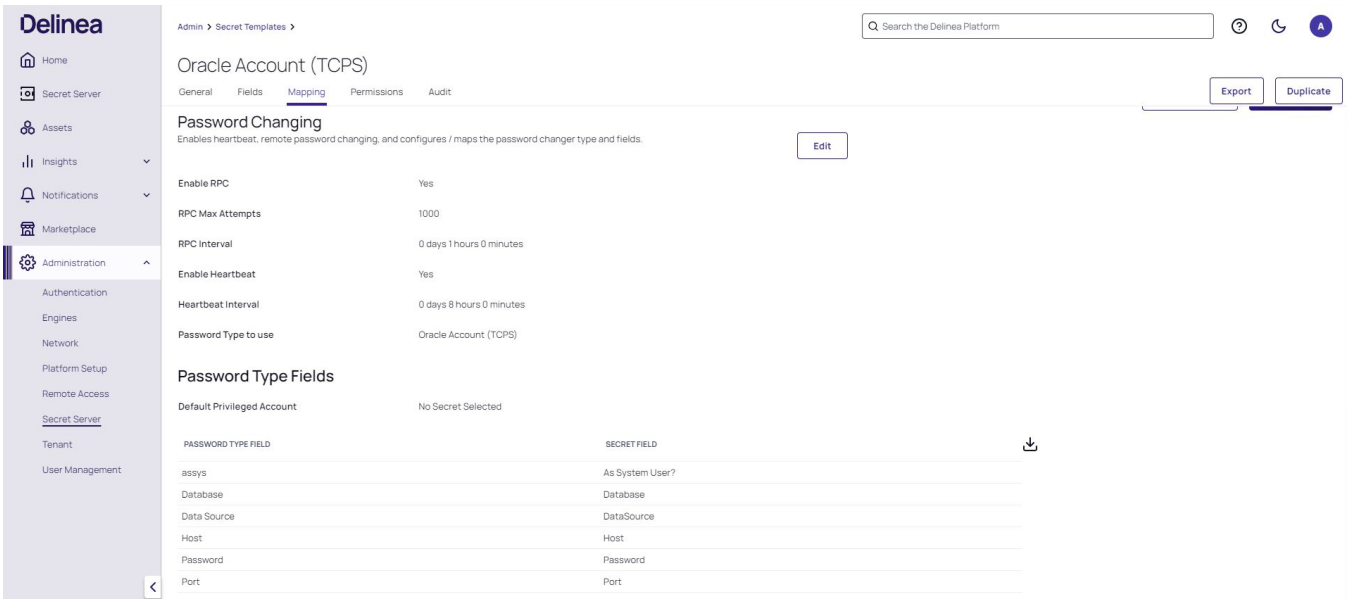
You can view and modify secret templates in the Secret Server administration panel. See "Creating or Editing Secret Templates" on page 1171 for more on the available options. Ensure that the secret template is in active status. See "Activating and Deactivating Templates" on page 1160 for details.

To navigate to an Oracle Account (TCPS) secret template:

1. Go to **Administration > Secret Secret Server**. The Secrets Administration page is displayed.
2. In the **Core Actions** section, click **Secret Templates**. The list of available templates is displayed.
3. Select an Oracle Account (TCPS) secret template and then click the **Mapping** tab.

You can check what secret template conforms to the selected RPC. The screenshot below shows that the Oracle Account (TCPS) RPC conforms to the identically titled secret template. It is possible to assign several password changers to one secret template. For more information, see "Assigning a Password Changer to a Secret Template" on page 916.

Secret templates determine the fields, launchers, and the remote password changer for secrets. To use the Oracle Account (TCPS) template on a secret, see "Managing Secrets" on page 1340.



Oracle RPC Templates

Secret Server now has four Oracle templates, three current and one legacy:

- Oracle Account
- Oracle Account (TCPS)
- Oracle Account (Template Ver 2)
- Oracle Account (Walletless)

See "Configuring Oracle Secret Templates" on page 1205 for details.

RPC for Postgres SQL

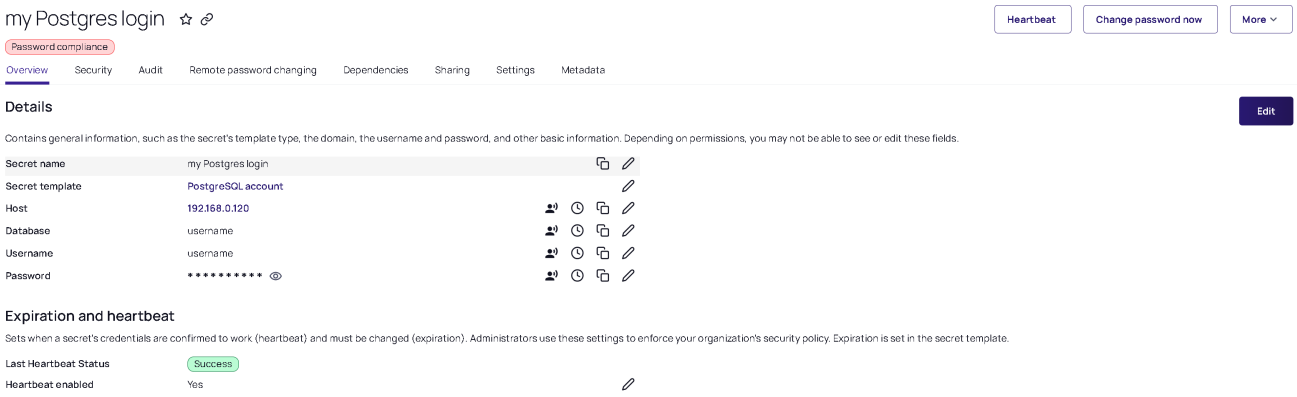
> This guide is for users familiar with PostgreSQL software who have it already installed and configured. If that is not the case, see PostgreSQL Downloads - Packages and installers for details on installing and configuring PostgreSQL.

Prerequisites

- A distributed engine.
  - Go to **Settings > Sites and engines**. You will see sites and their installed distributed engines here, along with any engines that are pending.
  - If you do not have a site with a distributed engine, see "Downloading and Installing a Distributed Engine " on page 760

Configuration

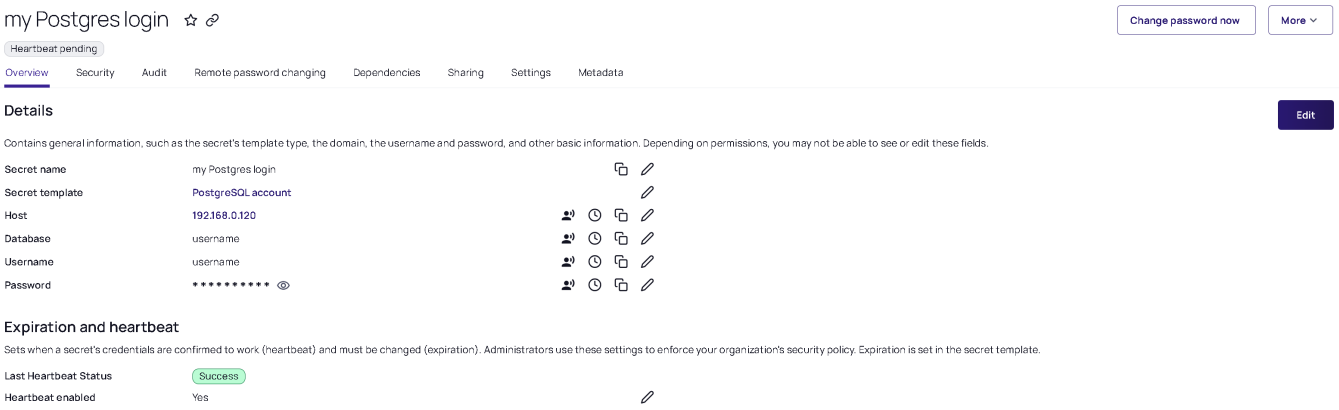
1. Log into Secret Server.
2. Navigate to **Secrets > All secrets**.
3. Select the **Create secret** button. The **Create new secret** page opens.
4. Search for and select the **PostgreSQL account** template. The popup refreshes automatically to reflect the necessary fields.
5. Complete the following fields:
   - **Secret name**: give the secret an appropriate name.
   - **Host**: the URL of the PostgreSQL tenant.
   - **Database**: specify the database on the PostgreSQL server to connect to.
   - **Username**: the username of the PostgreSQL account.
   - **Password**: the password of the PostgreSQL account.
   - **Site**: set a site with a distributed engine, as mentioned in the prerequisites.
   - **Auto Change Enabled**: leave unchecked.
6. Click **Create secret**. The newly created secret loads automatically for viewing:



Secret Server automatically runs a **Heartbeat** to verify if the credentials are valid, causing the status to change from **Pending** to **Success**. If the credentials are not valid the status will change from **Pending** to **Failed**.

> To verify the status of the heartbeat process immediately, navigate to **Settings > Heartbeat Log**.

You can change the password by selecting the **Change password now** option from the top right. This causes Secret Server to connect to the PostgreSQL instance specified in the secret and change the password in PostgreSQL, while simultaneously updating the password in the Secret as well.



RPC for Service Accounts

RPC can be performed on service accounts where the dependent services is automatically updated and restarted as the service account password is changed. Administrators are notified if a dependency fails to restart. The supported dependency types are IIS application pools, IIS application pool recycle, scheduled tasks, windows services, passwords embedded in .ini, .config, and other text files. Custom dependencies can be created using SSH, PowerShell, or SQL scripts. The application pool recycle only recycles the specified application pool, it does not update the password of the service account running the application pool. Secret Server attempts to unlock the service account should the account become locked during the dependency password change if there is a privileged account assigned to the secret.

Minimum Requirements for Windows Local Accounts

Due to a security issue (MS KB3178465), we do not allow Windows local accounts to change their own passwords unless the local admin account comes with the operating system. Other local admin accounts can also change their own passwords if the local security policy allows this. We recommend using the discovery privileged account to change these passwords. Each privileged account should meet the following requirements:

- Must be a domain user
- Must be a member of the local administrator group on all target end points

> The discovery account for Secret Server can also be used for RPC.

To use RPC, a specific registry setting is required:

- **Key:** HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
- **Value:** LocalAccountTokenFilterPolicy = 1

This setting is required to bypass Remote UAC restrictions.

Salesforce.com Password Changer

As of version 8.6, Secret Server supports password changing for Salesforce.com accounts under the Web User Account secret Template, by selecting the Salesforce Password Type under Remote Password Changing. In

version 11.7.3, and Distributed Engine version 8.4.27.0, the standalone Salesforce secret template was added.

The Secret Server Web server's outbound IP Address must be added to the IP Address white list for your Saleforce.com organization. Please refer to the Salesforce.com documentation for instructions on how to set this up. See Restricting Login IP Ranges for Your Organization.

In cases where this is not set up correctly, you may see the follow error in the Remote Password Changing logs:

`Login failed: LOGIN_MUST_USE_SECURITY_TOKEN: Invalid username, password, security token; or user locked out.`

Please note:

- Secret Server can only communicate to the following Salesforce default Login URLs: https://test.salesforce.com and https://login.salesforce.com.
- Having the domain URL in the secret will not work and will throw this exception: `Login failed: INVALID_LOGIN: Invalid username, password, security token; or user locked out.` Only those two URLs work.
- There are three required Salesforce configurations:
  - Go to **Setup > Administration > Users > Profile**. Choose the user profile. Make sure that **Enabled API** is checked. This option is not available in all versions of Salesforce. Other versions will not have this enabled by default. Please see this "Enable API" not available article. If this setting is not enabled in Salesforce you will get one of these errors: `ERROR: Secret 'Salesforce Test' (Id = 1063) on Site 'EARTH' returned (LoginFailed). Exception: Login failed: API_DISABLED_FOR_ORG: API is not enabled for this Organization` or `Partner,System.Web.Services.Protocols.SoapException: API_DISABLED_FOR_ORG: API is not enabled for this Organization or Partner.`
  - Configure network access and allowlist the distribute engine or Secret Server IP address. If this is internal, use the public IP address.
  - Go to **Setup > Company Settings > My Domain**. Edit my domain settings and make sure that **Prevent login from https://login.salesforce.com** is unchecked.

SAP Heartbeat and Password Changing

You can enable Secret Server to perform heartbeat and change passwords on SAP accounts by following the procedures as indicated below.

First, create a new privileged SAP account administrator secret, typically for the SAP or DDIC account that is used to log on to SAP for administrative tasks. Select the **SAP Account** template and enter all required information to create the new SAP account administrator secret. By default, the **Instance Number** will be 00 and the **Client Number** will be 001.


the default **System ID** for SAP is also NSP.

Second, create the account you are planning to change. Follow the same method as before and enter the current account password in the password field.

Third, in your new SAP account administrator secret, set the privileged account on the "Remote Password" changing tab.

for an account to have its password changed, even a privileged account changing its own password, it requires permissions in SAP.

For Secret Server 8.8.000000 and Higher

Download SAP .Net Connector 3.x and install it using the following procedure:

1. Navigate to `service.sap.com/connectors`.
2. Enter your credentials for the SAP Marketplace.
3. Click on SAP Connector for Microsoft .NET.
4. Download the .NET 4.0 Option with the proper bitness for your application pool (64-bit mode for most customers).
5. Install the downloaded file.
6. Copy the sapnco.dll and sapnco_utils.dll files into the bin folder of your web application. For Distributed Engine, add these files to the installation folder. Please see How to create an ignore file for Distributed Engine upgrades for details.
7. Recycle the application pool.

Once these steps are complete, heartbeat and password changing should be working.

Accounts can change their own SAP passwords just once per day. This is a restriction in the SAP software that cannot be changed. If an account needs its password to change more than once a day, use a privileged account to perform the reset.

If performing a Heartbeat on an SAP Secret fails with the error, `Exception: PASSWORD_EXPIRED`, it most likely means an administrator has reset the SAP account's password, and the account must log in and change its own password in SAP.

If having trouble, please verify the secret template is set up properly. See Configuring SAP SNC Account Secret Templates for more details.

For Secret Server 8.7.000000 and Below

1. Change your Secret Server application pool to run in 32-bit mode.
2. Download the SAP GUI version 720 from the SAP Community website.
3. Extract the downloaded ZIP file. Depending on the version, the extracted download will have a GUI or Frontend Tools directory.
4. Copy that directory over to the machine running Secret Server.
5. Run the installer inside the directory. The install should take only a couple of minutes.

SQL Server RPC

Remote Password Changing (RPC) for SQL Server accounts in Secret Server allows administrators to automate the management of SQL Server account passwords, ensuring they remain synchronized with Secret Server policies. This feature supports the use of privileged accounts to change SQL Server passwords without knowing the current password, enhancing security and efficiency. Administrators can create SQL Server accounts, assign the necessary permissions, and configure Secret Server to use these privileged accounts for password changes. The SQL Server Account template in Secret Server is preconfigured to facilitate this process, and administrators can customize it as needed. This functionality ensures that SQL Server account passwords are automatically updated according to defined schedules or upon expiration, maintaining compliance with security policies and reducing the risk of unauthorized access.

Creating and Using a SQL Server Privileged Account

Overview

This document enables a user to password change SQL accounts using a privileged account. Enabling the takeover of those accounts without knowing their password.

Procedure

Task 1: Creating an Account

1. Open SQL Server Management Studio and connect to your database server.
2. Expand the root level security folder.
3. Right click on the **Logins** folder and select **New Login**.
4. Type the account's login name in the **Login Name** text box.
5. Click to select the **SQL Authentication** selection button.
6. Go to Secret Server.
7. Create a secret using the **SQL Server Account** template.
8. Give it the same username as the login name you just created.
9. For best security, click the **Generate** button on the secret password field and copy that password to the account creation wizard in SQL Server Management Studio.
10. Click **OK** button to save your secret.

Task 2: Assigning Permissions

1. Return to SQL Server Management Studio and connect to your database server.
2. Right click the SQL login and click **Properties**.
3. Select **Securables** in the left column.
4. In the **Permissions** table on the **Explicit** tab, click to select the **Grant** check box for the **Alter any login** row.
5. Click the **OK** button.

Step 3: Using the Account

1. In Secret Server, select the SQL account secret for your new privileged account.
2. Select the **Remote Password Changing** tab.
3. Click the **Edit** button.
4. Click to select **Privileged Account Credentials** on the **Change Password Using** selection button.
5. Click the **No Selected Secret** link. The Select a Secret popup appears.
6. Locate and select the secret you created earlier in the folder tree.
7. Click the **Save** button. The popup disappears.
8. Click the **Change Password Remotely** button.
9. Provide or generate a new password.
10. Click the **Change** button. You have now successfully changed a SQL account password using a privileged account.

> You can also assign an account for multiple secrets by creating a secret policy and applying that policy to a folder.

RPC on SQL Server Accounts

Overview

This address using a Secret Server privileged account to change SQL Server accounts. This enables taking over those accounts without knowing their password.

Creating the Account

1. Open SQL Server Management Studio.
2. Connect to your database server.
3. Expand the root-level security folder.
4. Right-click the **Logins** folder and select **New Login**.
5. Give the account a log on name.
6. Select SQL authentication.
7. Go to Secret Server.
8. Create a secret using the **SQL Server Account** template.
9. Assign it the desired username .
10. Click the **Generate** button on the secret password field to create a password.
11. Copy that password to the account creation wizard in SQL Server Management Studio.
12. Click the **OK** button to save the secret.

Assign Permissions

1. In SQL Server Management Studio, go to **Security > Logins** in the object explorer.
2. Right click on the SQL login object and select **Properties**. The Login Properties dialog box appears.
3. Select **Securables** in the **Select a page** list.
4. Find the **Alter any login** permission on the **Explicit** tab at the bottom of the dialog box.
5. Click to select the **Grant** check box for that permission.
6. Click the **OK** button.
7. Similarly, enable the **Control Server** permission. This is for changing the target logins that are members of the **sysadmin** fixed server role or grantees of this permission.

Using the Account

1. In Secret Server, open the SQL Server secret that you created.
2. Click the **Remote Password Changing** tab.
3. Click the **Edit** link.
4. Click to select **Privileged Account Credentials** in the **Change Password Using** selection buttons. The Privileged Account section appears.
5. Click the **No Secret Selected** link.
6. Select the secret you created earlier. The secret appears in the Privileged Account section.
7. Click the **Save** button.
8. Click the **Change password remotely** button.
9. Provide or generate a new password.
10. Click the **Change** button. You have now successfully changed a SQL Server account password using a privileged account.

> You can also assign the account for use by multiple secrets by creating a secret policy and applying that policy to a folder.

Using SQL Privileged Account for RPC

Overview

This document enables you to password change SQL accounts using a privileged account. This allows you to take over accounts without knowing their password.

Task 1: Creating an Account

1. Open SQL Server Management Studio.
2. Connect to your database server.
3. Expand the root-level security folder.

4. Right click the **Logins** folder.
5. Click **New login**.
6. Name the account.
7. Click **SQL Authentication**.
8. Go to Secret Server.
9. Create a secret using the **SQL Server Account** template. Give it the same username as the login name you are creating.
10. For best security, click the **Generate** button on the secret password field.
11. Copy the generated password to the account creation wizard in SQL Server Management Studio.
12. Click **OK**.
13. Save your secret.

Task 2: Assigning Permissions

1. Right click on the SQL login and click **Properties**.
2. Select **Securables** in the left column.
3. Select **Grant** for **Alter any login**.

Task 3: Using the Account

1. In Secret Server, select the SQL account secret you are going to have represent your new privileged account.
2. Select the **Remote Password Changing** tab and click **Edit**.
3. Click the **Change Password Using** selection button and select **Privilege Account Credentials**.
4. Click the **No Selected Secret** link.
5. Find and select the secret created for the privileged account in the first task.
6. Click the **Save** button.
7. Click the **Change password remotely** button.
8. Provide or generate a new password.
9. Click the **Change** button. You have now successfully changed a SQL account password using a privileged account.


You can also assign the account for use by multiple secrets by creating a secret policy and applying that policy to a folder.

RPC for SSH

Secret Server's Remote Password Changing (RPC) feature for SSH allows administrators to automatically update passwords for remote accounts using SSH protocols. This functionality ensures that passwords are rotated according to defined schedules or upon expiration, enhancing security by maintaining strong, up-to-date credentials. Secret Server supports SSH key rotation, enabling the management of private SSH keys for tasks such

as PuTTY launcher sessions and Unix/Linux discovery. Using SSH RPC, organizations can automate password changes, enforce password policies, and reduce the risk of credential-based attacks.

Creating a Custom SSH Password Changer

To create a custom SSH password changer for a device that supports changing passwords via SSH, follow the procedure for "Creating a Custom Password Changer" on page 905. Be sure to choose a base password changer with a name that ends in `(SSH)`.

Editing Custom Commands

The SSH type changers use the SSH protocol to access the machine. This type contains custom commands for password reset and can contain commands for the verify password functionality but most SSH type changers simply verify that a connection can be established with the username and password. The Telnet type changers use the Telnet protocol in order to access the machine and contain custom commands for both the password reset and the verify password functionality. The verify functionality is used in the heartbeat, as well as verifying that the password was changed successfully.

SSH key rotation type changers also include post-reset success and failure custom commands. These extra command sets are run after both the reset and verify functions are run and are used to either finalize the key rotation and password change (success) or clean up after a failure. If both the reset and verify functions are successful, the post-reset success command set is run. If either the reset or the verify fail, the post-reset failure command set is run.

To edit the custom commands, click on the **Edit** Commands button. This sets the command grids into Edit mode where you can add, update, or delete the commands in order to suit their purpose.

RPC-Mapped Text-Entry Fields

Prepend a `$` to any text-entry field name to access that field. For example, to echo the notes value for a secret, you would use this command: `echo $Notes`. Commonly accessed fields include:

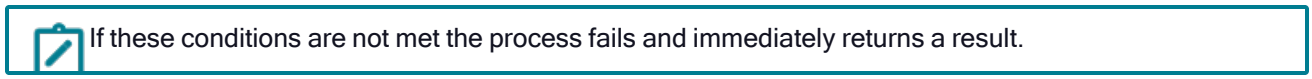
- `$USERNAME` The username text-entry field mapped in RPC on the secret template.
- `$CURRENTPASSWORD` The password text-entry field mapped in RPC on the secret template.
- `$NEWPASSWORD` The next password (filled in Next Password textbox or auto-generated).
- `$PRIVATEKEY` The private key text-entry field mapped in RPC on the secret template.
- `$NEWPRIVATEKEY` The next private key (filled in Next Private Key text box or auto-generated).
- `$CURRENTPUBLICKEY` The public key text-entry field mapped in RPC on the secret template.
- `$NEWPUBLICKEY` The next public key (generated from the next private key).
- `$PASSPHRASE` The passphrase text-entry field mapped in RPC on the secret template.
- `$NEWPASSPHRASE` The next passphrase (filled in Next Private Key Passphrase text box or auto-generated).

Associated Reset Secrets

- `$[1]$` Adding this prefix to any text-entry field targets the associated reset secret with order 1.
- `$[1]$USERNAME` The mapped username of the associated secret, identified by order. Can also reference any other property on the associated secret. Common examples include:
- `$[1]$PASSWORD`
- `$[1]$CURRENTPASSWORD`
- `$[1]$PRIVATE KEY`
- `$[1]$PRIVATE KEY PASSPHRASE`
- `$[SID:105]` Adding this prefix to any text-entry field targets the associated reset secret with a secret Id of 105.
- `$[SID:105]$USERNAME` The mapped username of the associated secret, identified by secret id. Like referencing an associated secret by order, referencing by secret id can also access any text-entry field on the secret by name.

Check-Result Commands

- `$$CHECKCONTAINS <text>` Checks that the response from last command contains `<text>`.
- `$$CHECKFOR <text>` Checks that the response from the last command equals `<text>`.
- `$$CHECKNOTCONTAINS <text>` Checks that the response from last command does not contain `<text>`.

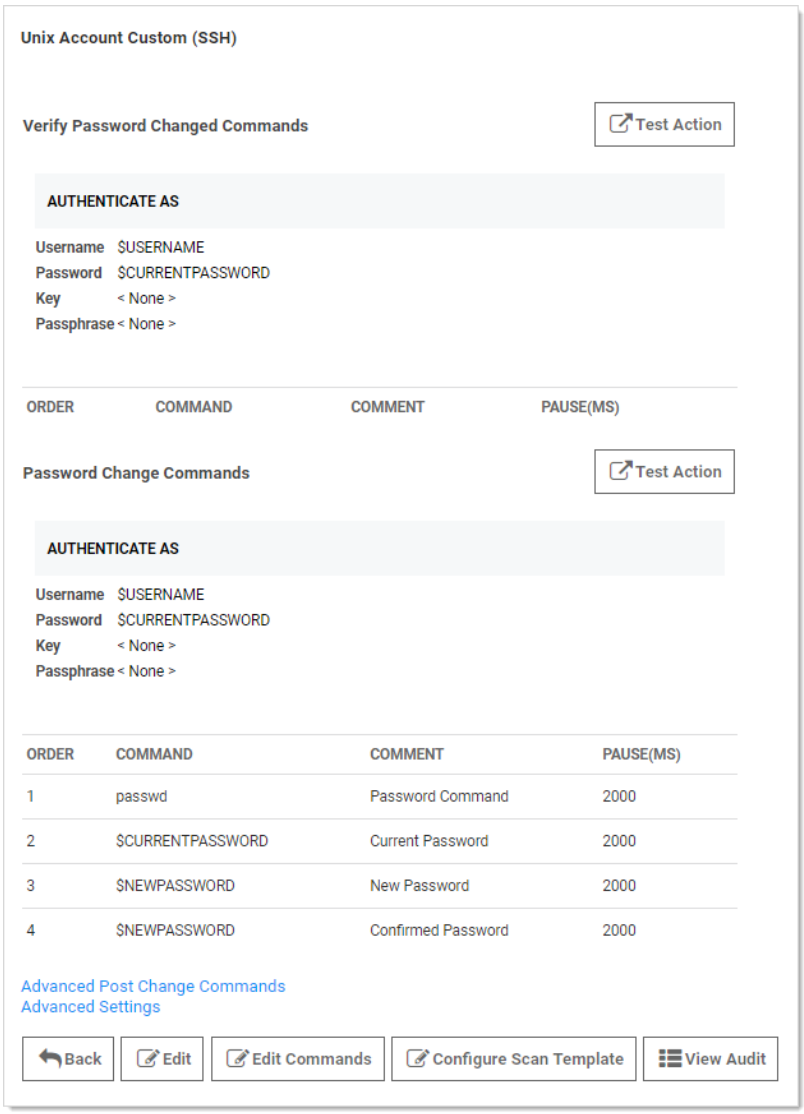


If you want to exit out of the command set early without triggering a failure, echo an "OK" on the line immediately preceding the `exit 0;` statement. "OK" must be the only text in the response from the server for this to work.

You can test out your password reset and verify password command sets by clicking on the **Test Action** buttons next to the relevant sections. All communication between Secret Server and the target machine is displayed when using these test buttons.

Mapping an SSH Key or Private Key Passphrase for Authentication

Some password changers may be customized to use SSH key authentication. Secret Server needs to know which text-entry fields contain the key and the passphrase. These text-entry fields can be specified after clicking **Edit** from the password changer page.



The key and passphrase must be identified by a $ sign and the secret text-entry field name, which can be obtained from the secret template.

To set which text-entry fields are your key and passphrase, go to the extended mappings for a secret template by clicking **Extended Mappings** from the **Secret Template Edit** page. Select the text-entry fields that correspond to the SSH private key and passphrase if applicable. No matter what you name your key text-entry field, Secret Server knows what it is. This is set up by default, so you should not need to do this unless you've created custom Unix templates you want to use keys with.

Once Secret Server knows which text-entry fields contain the private key and private key passphrases, it can automatically use them as a part of launchers.

Remote Password Changing for Okta

When using Remote Password Changing (RPC), Secrets automatically change remote account passwords when they expire, either immediately or on a defined schedule. You can also configure password strength and other

attributes.

> If you need information on configuring Okta for SAML or assigning a password changer to an Okta secret template go to: "Configuring SAML Okta" on page 417 and "Okta Secret Template for RPC" on page 968.

Prerequisites

Before configuring the integration, ensure the following requirements are met:

1. An Okta administrator account is required to create the API token, which will be used for running both a heartbeat and RPC.
2. An Okta user account whose credentials will be verified (heartbeat) or changed (RPC).
3. The **Okta Verify** app, from the App Store or Google Play. See Configure Okta Verify for setup details.
4. If using **Okta password changers with a distributed engine**, ensure the version is 8.4.32.0 or newer.
5. A valid Okta API key, for performing the password change.

Setup

Creating a Privileged Account

1. In Okta, go to **Security > Administrator Roles** and select **Add Administrator Role**.
2. The minimum role permission needed is **Organization Administrator**.
3. Go to the user's profile.
4. Select the **Assignments** tab.
5. Select **Assign** and choose the appropriate administrator role from the list.
6. Select **Save** to apply the administrator role to the user.

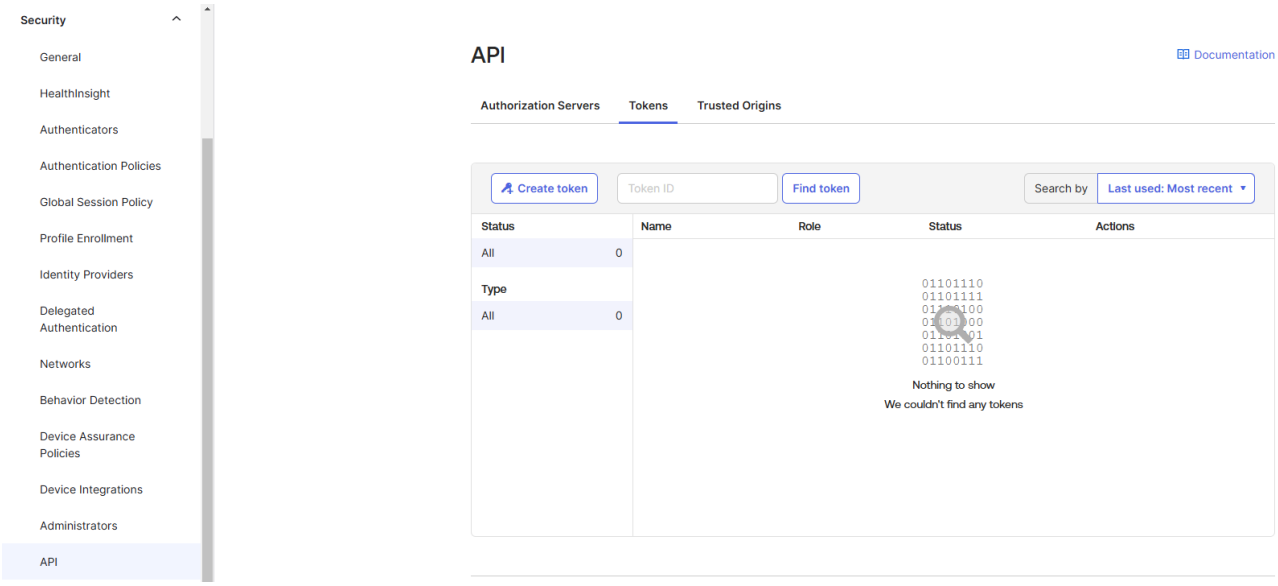
Creating a User Account

1. Log into the **Okta Administrator Dashboard** using your administrator credentials.
2. Navigate to **Directory > People**.
3. Select **Add Person** and complete the following fields in the popup dialog:
   - First name
   - Last name
   - Primary e-mail
   - Username
4. Select **Save** to create the new user account.

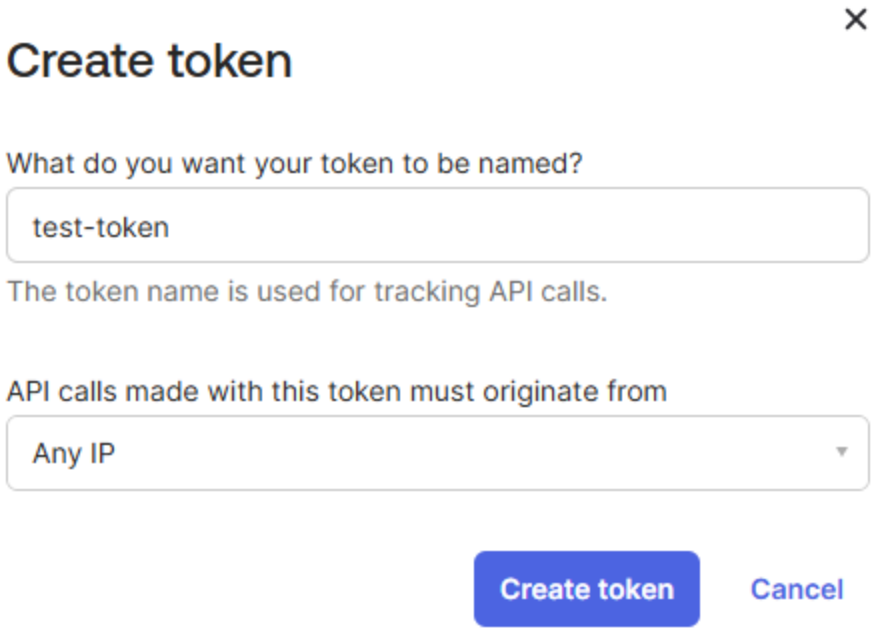
Generating an API Key

Heartbeat and RPC operations are performed through API calls. The Okta Secret must be configured with a separate API Key Secret that contains the necessary API credentials for authentication. This API key must be generated:

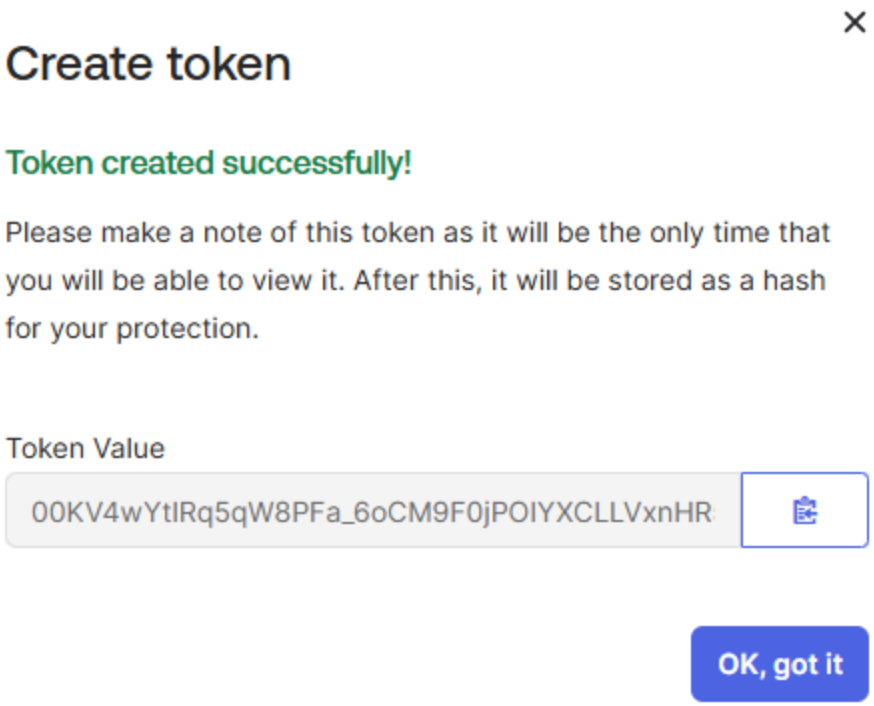
1. Log into Okta with an Admin account.
2. Navigate to **Security > API**.
3. In the **Tokens** tab, select the **Create Token** button. The Create token popup appears:



4. Provide a name for the token, select **Any IP** from the dropdown, and click **Create Token**:



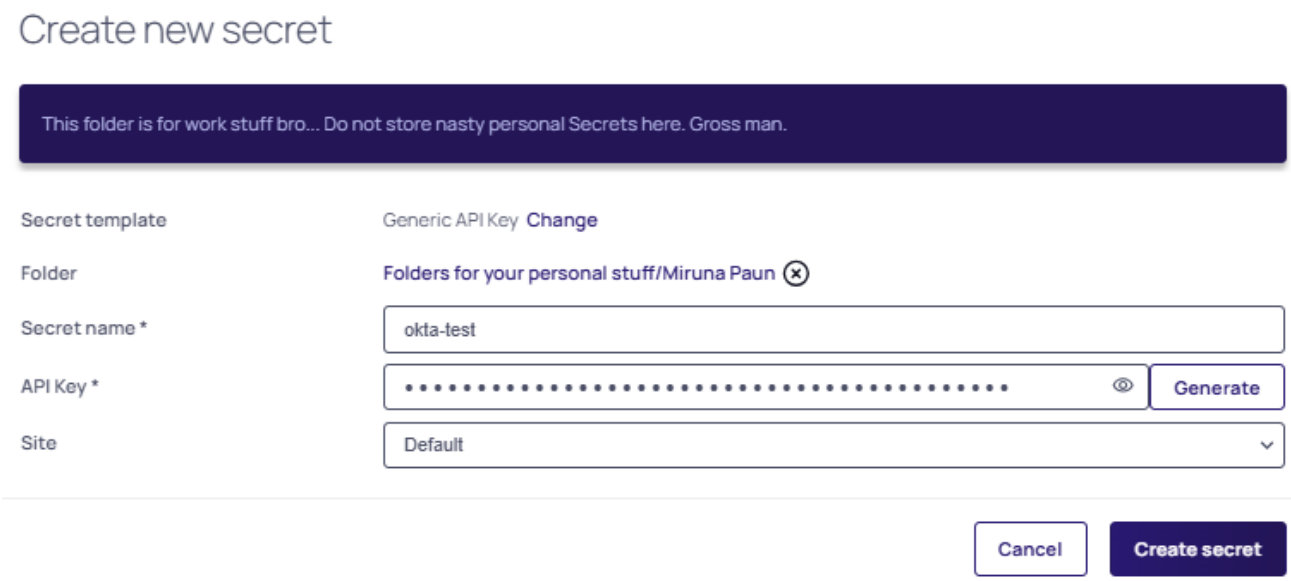
5. A success message for the created token appears. Copy the generated API key (Token Value) to a secure location:



Configuring Okta RPC in secret-server

Create a secret to hold the Okta API token

1. Log into secret-server.
2. Navigate to **Secret Server > All secrets**.
3. Select the **Create secret** button. The **Create new secret** popup appears.
4. Search for and select the **Generic API Key** template.
5. Create a Secret of type **Generic API Key** to hold the Okta token value:



6. In the **API key** field, paste the Token Value (API Key) copied from the Okta interface.
7. Select the **Create secret** button to save the secret. The newly created secret opens by default.

Create a secret to hold the Okta user account

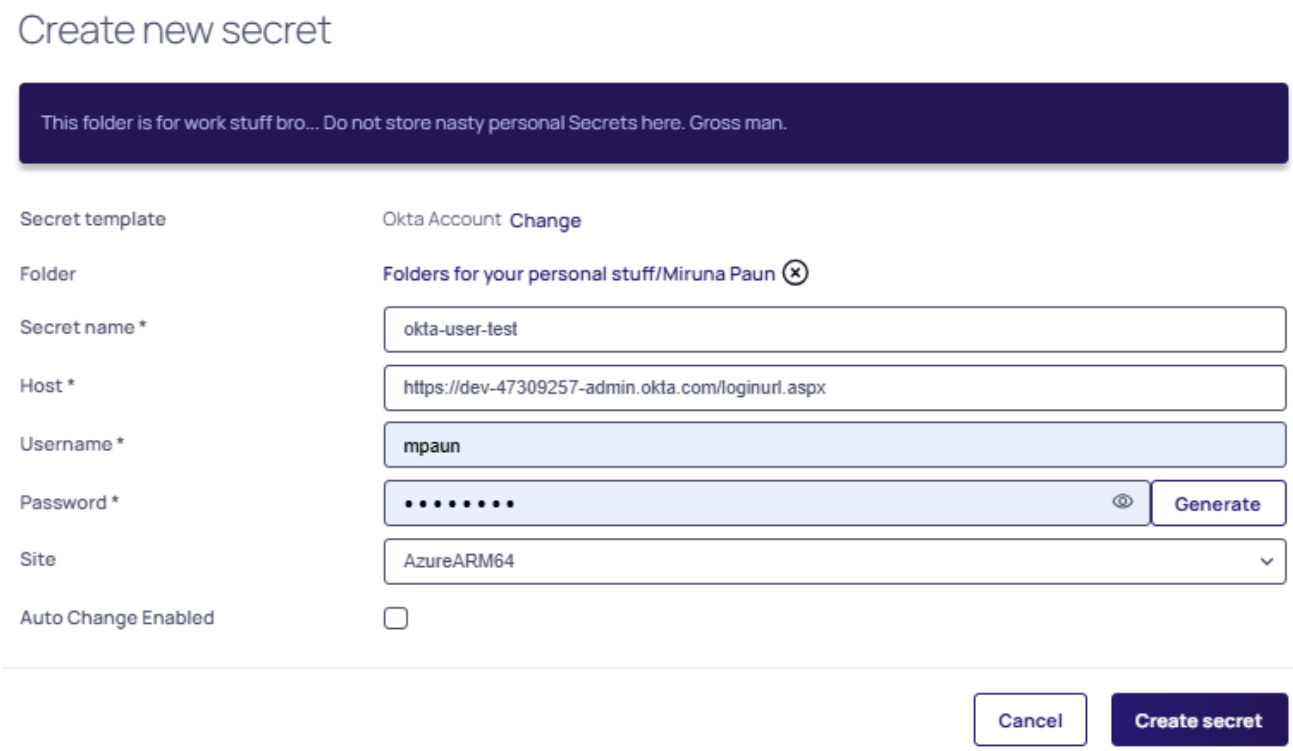
This account will have its credentials verified or password changed:

1. Log into secret-server.
2. Navigate to **Secret Server > All secrets**.
3. Select the **Create secret** button. The **Create new secret** page opens.
4. Search for and select the **Okta Account** template.
5. Complete the following fields:
   a. **Secret name**: Provide a name for your secret.
   b. **Host**: Use your tenant URL (e.g. https://yourcompany.okta.com).

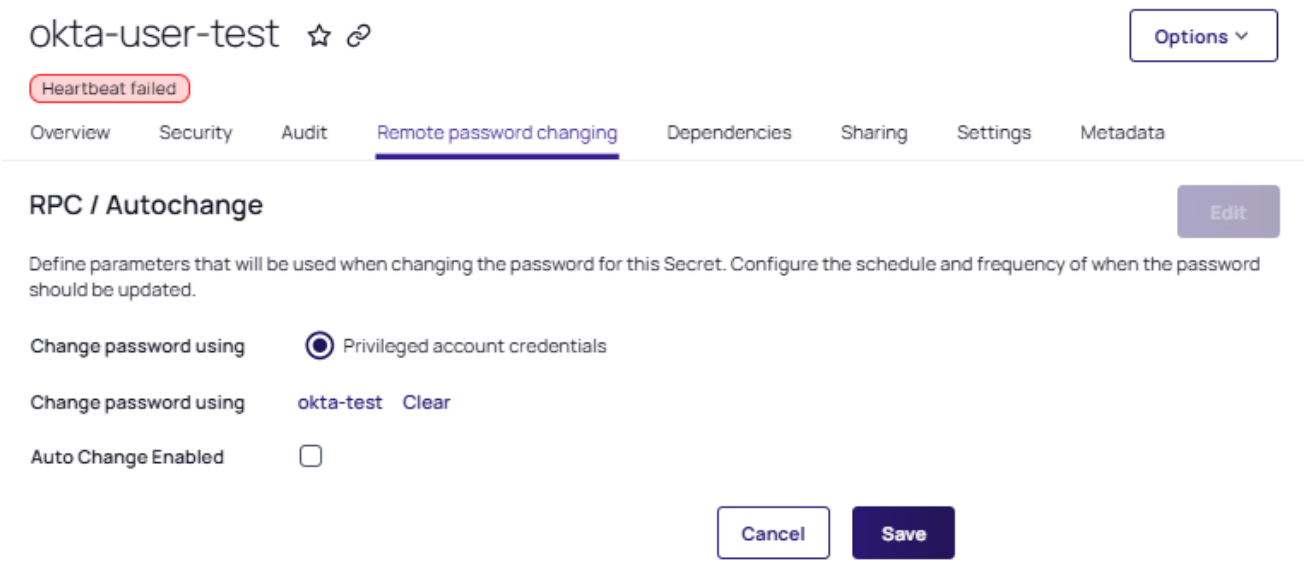
c. **Username**: Enter the username of your Okta account.

d. **Password**: Enter the password of your Okta account.

6. In the **Site** field, set a site with a distributed engine connected to the internet:



7. Select the **Create secret** button to save the secret. The newly created secret opens by default.
8. Access the **Remote password changing** tab.
9. Select **Edit** in the **RPC / Autochange** section.
10. Select the option for **Change password using** titled **Privileged account credentials**.
11. A second option for **Change password using** appears, click on **No secret selected**.
12. Search for the secret you created previously that holds the API key of your Okta account:



13. Click **Save** to implement your changes, and return to the **Overview** tab.
14. Select the **Heartbeat** button in the top right corner, to validate the credentials. If the credentials are valid the status will change from **Pending** to **Success**.

> To check the heartbeat status immediately, go to **Settings > Heartbeat Log**.

All Okta accounts require an admin API key in the privileged account field under the **Remote Password Changing** tab to change or verify passwords. If no admin API key is assigned, a warning notification appears:

> WARNING
> An API Key application account is required to change or verify the password on this account. Dismiss

To maintain security, rotate the API key regularly. Okta recommends rotating the API keys at least every 90 days or as per your organization's security policies.

Verification

To monitor heartbeat status, go to **Settings > Heartbeat Log**. To change the password immediately, select **Change password now**.

RPC for Snowflake in Secret Server

RPC for Snowflake in Secret Server applies to Snowflake SQL database user accounts, including both admin and non-admin user accounts.

Prerequisites

Make sure you have:

- Two active Snowflake accounts. One of these accounts must be a privileged admin account which will be used for password changing.
- A Secret Server user which can create two snowflake secrets.
  - Optionally, admin credentials for Secret Server (the **ACCOUNTADMIN** role must be assigned to the admin account).
- The RPC feature enabled in Secret Server.
- Permission to create and configure secrets.
- Heartbeat monitoring and remote password-changing features enabled on Secret Server.
- A site with a distributed engine which has access to the internet.

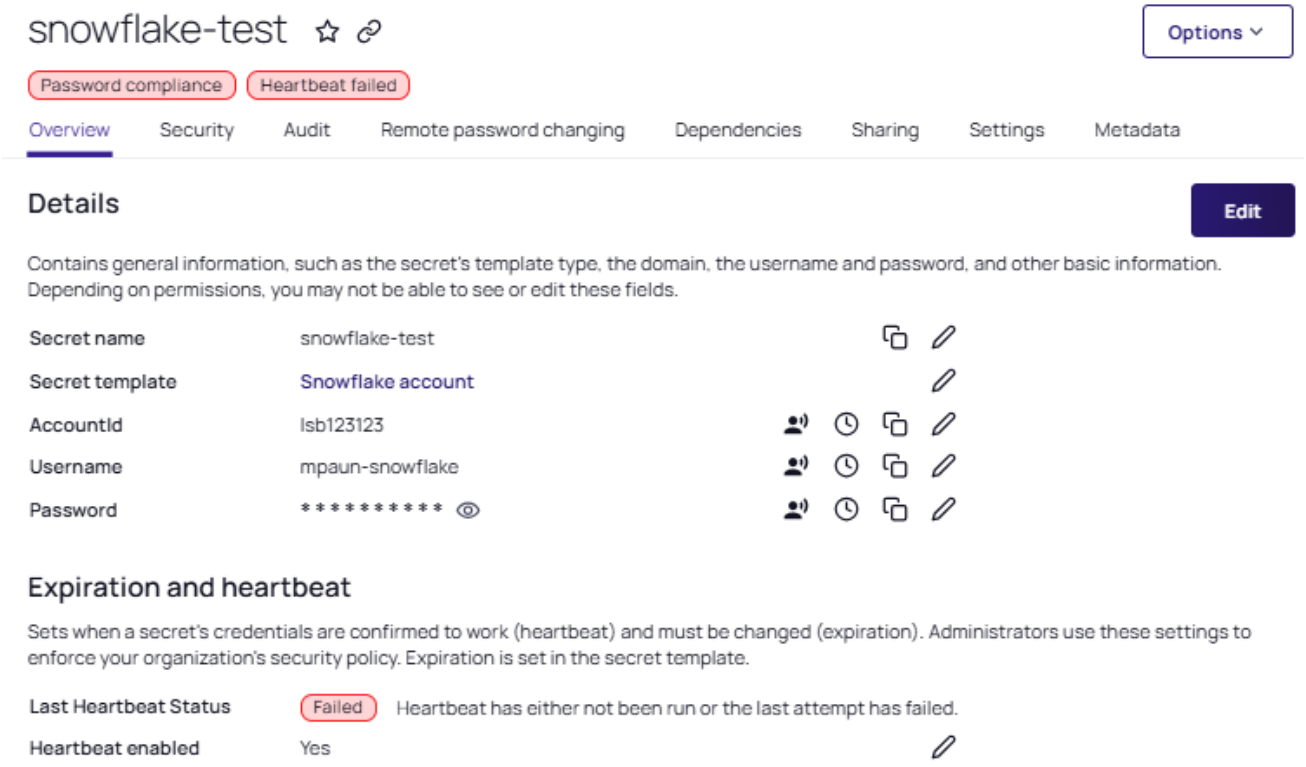
Configuration

1. Log into Secret Server.
2. Navigate to **Secrets >All secrets** and click the **Create secret** button, the **Create new secret** popup appears.
3. Search for the **Snowflake account** template and select it. The popup refreshes automatically to reflect the fields you must fill in.
4. Complete the following fields:
   a. **Secret name**: give the secret an appropriate name.
   b. **AccountId**: you will find this as a part of your Snowflake URL (starts with lsb followed by several numbers).
   c. **Username**: the username used to sign into the Snowflake account.
   d. **Password**: the password used to sign into the Snowflake account.
   e. **Site**: set a site with a distributed engine that can access Snowflake services.
   f. Leave **Auto Change Enabled** unchecked and click **Create secret**. The newly created secret loads automatically for viewing.
5. The **Heartbeat** operation runs automatically to check if the entered credentials are valid. If the credentials are valid the status will change from **Pending** to **Success**.

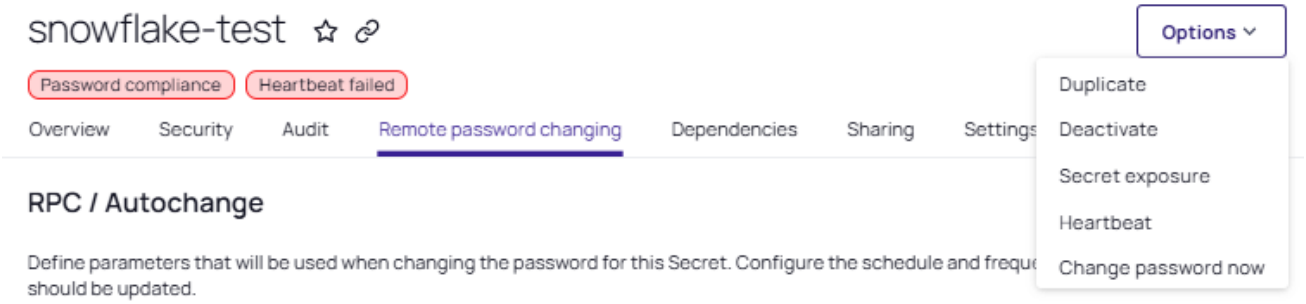
   If the credentials are not valid the status will change from **Pending** to **Failed**.

The distributed engine checks for RPC every 300 seconds. If the heartbeat state remains in **Pending** for longer than 300 seconds, confirm that the site has an operational distributed engine by accessing **Settings > Sites and engines**.

To verify the status of the heartbeat processes, navigate to **Settings > Heartbeat Log**.



6. Navigate to the **Remote password changing** tab and select **Edit** for the **RPC/Autochange** section.
7. For **Change password using**, select the **Privileged account credentials** option.
   a. If you chose the option above, the **Change password using** option appears, and you must select a secret by clicking on the **No secret selected** link. A popup will appear where you can search for the secret you want to associate. Select a Snowflake user with the **ADMINACCOUNT** role used to process the password change.
   b. Click **Save**.
8. (Optional) Access the **Change password now** option button from the top right corner if you want to change the secret password. Alternatively, it can be found under the **Options** dropdown list:



Heartbeat Overview

Heartbeat, which can be integrated with RPC, allows Secret Server to verify if the credentials stored in a secret can successfully authenticate with the target system. This ensures that the credentials are still valid and have not been

changed outside of Secret Server.

You can configure "Event Pipelines" on page 285 to track whether an RPC has failed. Heartbeats allow you to check whether a password is incorrect and the machine is online.

If a guest account exists on the domain, an Active Directory secret's heartbeat will mistakenly report success. Microsoft disables the guest account by default for security reasons.

Here are the key aspects of heartbeat:

Automatic Credential Testing

Heartbeats allow secrets to have their credentials tested automatically to ensure they are accurate and up-to-date. This helps in managing secrets and preventing them from being out of sync.

SMB Fallback

- To maximize compatibility across different versions of Windows, Secret Server can make a second attempt to use the secret via SMB if the initial heartbeat fails.
- This fallback can be enabled or disabled based on the requirement.

Secret Server makes a second attempt to use the Secret via SMB when **Use SMB heartbeat fallback** is checked. When **Use SMB heartbeat fallback** is not selected this second attempt will not be made.

Heartbeat Flexibility and Useability

- By default, heartbeat is turned off in Secret Server.
- Administrators can enable heartbeat for specific secrets and run it manually if needed.
- The status of the last heartbeat run is displayed in the secret's details, and administrators can manually trigger a heartbeat check from the Secret View page.

Heartbeat Status Codes

See "Heartbeat Status Codes" on page 1045 for details.

- Success: Successful authentication.
- Failed: Unsuccessful authentication.
- Unable to Connect: Unsuccessful connection with target machine.
- Unknown Error: Unknown error–see the heartbeat log.

Failure Response

- If a heartbeat fails, the secret is flagged as "heartbeat failed" and will not be checked again until the secret items are edited by a user.
- If the target machine is unavailable, the secret is flagged as "heartbeat unable to connect" and will continue to be checked at the next interval.

Configuring Heartbeat

Heartbeat is configured from the secret template designer. The heartbeat interval determines how often the secret credentials are tested.

Enabling Heartbeat in RPC

To enable heartbeat, ensure it is enabled on the **Remote Password Changing Configuration** page:

1. Navigate to **Admin > Remote Password Changing**.
2. Click the **Edit** button.
3. Click to select the **Enable Heartbeat** check box.
4. Click the **Save** button.

> Heartbeat must also be enabled on the secret template by setting the **Enable Remote Password Changing Heartbeat** setting.

Heartbeat Failure Alert Notification

Secret Server provides a feature to alert users when a heartbeat check fails for any secret. This ensures that administrators and users are promptly informed about potential issues with credential accuracy or connectivity. Here are the key aspects of configuring and receiving heartbeat failure alerts:

Enabling Email Alerts for Heartbeat Failures

- On the Preferences page, you can enable the setting "Send Email Alerts when Heartbeat Fails for Secrets" to receive email notifications whenever a heartbeat fails for any secret that the user has view access to.
- This setting ensures that users are immediately informed about any issues with the credentials stored in Secret Server.

Personalize Settings

In the Secret Settings tab, users can configure specific email notification settings:

- Send Email When Heartbeat Fails: This option sends an email to the user when a heartbeat function fails for the secret. The email contains the secret name, error code, and details about the failure.
- These settings are user-specific and do not apply to other users who have view, edit, or owner permissions for the secret.

Event Subscriptions

- Secret Server's Event Subscriptions feature allows administrators to set up customizable alerts that send email notifications when specified actions are performed or events occur, such as a heartbeat failure.
- This feature can be configured to alert users or administrators about heartbeat failures, ensuring timely awareness and response to credential issues.

Heartbeat Log

The Heartbeat Log provides a detailed record of all heartbeat activities within Secret Server. Administrators can review this log to monitor the health and accessibility of secrets, ensuring that credentials are valid and services are operational.

Heartbeat Logs

The heartbeat logs for a specific secret can be accessed by clicking the **View Audit** button on the **Secret View** page and clicking to enable the **Display Password Changing Log** check box. The heartbeat logs for all secrets can be accessed by navigating to **Administration > Remote Password Changing** and scrolling down to the second set of logs.

Heartbeat Status Codes

- **AccessDenied**: Account does not have the rights to log into the resource. Example: Remote login is not enabled for a Windows local account.
- **AccountLockedOut**: Account is locked out in the domain or on the workstation for Windows local accounts or Linux accounts.
- **ArgumentError**: Incorrect arguments have been provided to complete the Heartbeat. Example: Trying to use the new Entra ID secret template without a privileged secret mapped.
- **Disabled**: Heartbeat is disabled because the secret used QuantumLock, does not exist, is disabled, or does not have the correct license level activated.
- **DnsMismatch**: secret-server connected to an unexpected host when trying to complete a heartbeat, likely a DNS or network problem.
- **Failed**: The credentials are either incorrect or the account does not have permission to log in.
- **Failed Unknown**: Catch-all for any responses we don't recognize.
- **IncompatibleHost**: An incompatible function was applied to the device or source. Example: Trying to use a Linux password changer for a Windows account.
- **NeedsImmediateRetry**: The heartbeat feature uses PowerShell, and the MaxShellsPerUser amount was exceeded and will be tried again.
- **Pending**: The secret is set to be processed during the next batch of processing.
- **PrivilegedAccountRequired**: Secrets that require a pprivileged account to run a heartbeat, but the account was not linked or was missing when the heartbeat was queued.

- **Processing**: The heartbeat was sent to the engine for processing and secret-server is awaiting a response.
- **Success**: Successful credential validation.
- **UnableToConnect**: Secret Server was unable to contact the target system. Ensure that the domain, IP address, or hostname is correct and resolvable from the server that Secret Server is installed on.
- **UnableToValidateServerPublicKey**: The Linux key-based credentials are incorrect.
- **UnknownError**: Check the heartbeat log on the **Remote Password Changing** page for details, and contact Support for assistance. This error typically refers to other cases where we could not determine the reason for the failure but reached a resource such as Active Directory. Example: "User Name could not be found."

Please keep in mind the pipeline appears to log a heartbeat failure if it receives any status other than: "Success", "Pending", "Disabled", or "Unable to Connect". This means that "Processing" will also be treated as a failure.

> Enabling the built-in guest user in Active Directory can cause confusion because heartbeat returns a "success" status for non-existent accounts. To avoid this, disable the guest user when setting up AD.

Remote Accounts Supported

For the most up-to-date list of account types supported by RPC, see "Password Changer List" on page 909.

Running Heartbeat for a Secret

Heartbeat runs in a background thread to check each secret where it is enabled. If the credential test fails, the secret is flagged as heartbeat failed and out of sync. To avoid locking out the account, heartbeat no longer run on that secret until the secret items are edited by the user. If the machine is determined to be unavailable, the secret is flagged as heartbeat unable to connect and the secret continues to be checked on the heartbeat interval.

To manually use heartbeat to check the credentials, the **Secret View** page has a **Heartbeat Now** button. The button marks the password as heartbeat pending. The background thread processes the secret in the next 10 seconds, and when the page is refreshed the heartbeat status is updated.

> Heartbeat does not work on Windows accounts on the server that is running Secret Server. These accounts are flagged with an "Incompatible Host" status.

To run heartbeat for a secret:

1. From **Dashboard**, click the secret you would like to test.
2. Click the **View** button. The **Last Heartbeat** field of the secret shows the last date and time that Heartbeat ran for this secret.
3. To run Heartbeat once more, click **Run Heartbeat** at the bottom of the Secret.
4. Monitor the **Last Heartbeat** field to see the updated status. This may take a few seconds to complete.

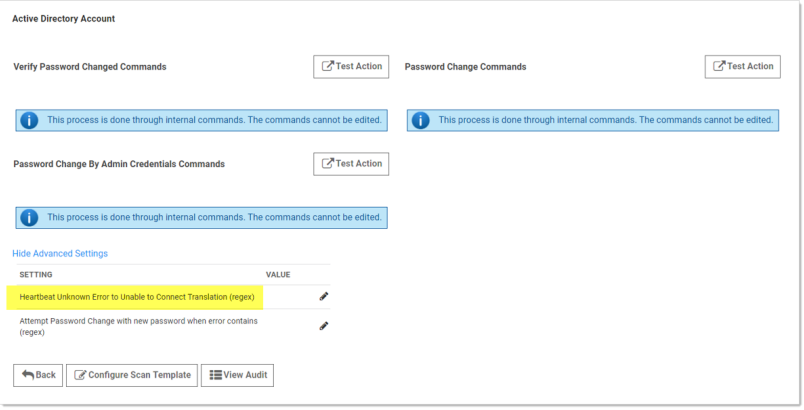
If you receive any Heartbeat status code aside from Success, you can check the Heartbeat log for details. To view the entry, Go to **Admin > Remote Password Changing** and then search for the secret name in the **Search** field of the **Heartbeat Log**.

Treating Heartbeat "Unknown Errors" as Connection Failures

> This setting was previously called "Password Change Error Code Translation (regex)."

The Secret Server "Heartbeat Unknown Error to Unable to Connect Translation (regex)" setting can translate UnknownError heartbeat errors into connection errors based on text, such as the error code, in the error message. Using a regular expression, which you define, Secret Server scans heartbeat UnknownError messages for specific text strings. When there is a match, Secret Server changes the UnknownError to an "Unable to Connect" heartbeat error. This setting is useful if a custom command error is interpreted as UnknownError but the message indicates it actually was unable to connect. The translated connection error will cause Secret Server to attempt another heartbeat.

Figure: Heartbeat Unknown Error to Unable to Connect Translation (regex) Setting



> The UnknownError error is very common when running scripts and commands, making the regex discrimination desirable.

Logic:

(RPC UnknownError) AND (Regex match in error message) => RPC status changed to "Unable to Connect"

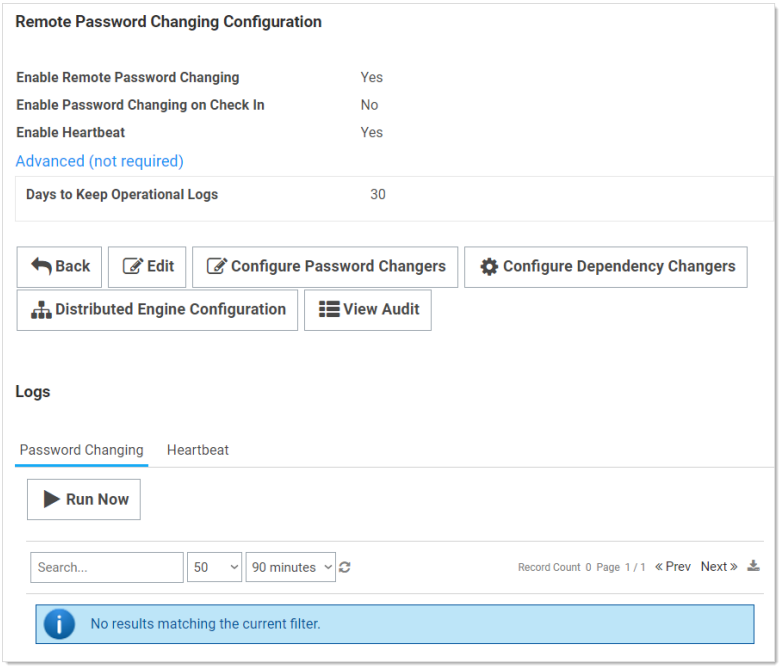
Example:

`.*error code is 10060.*` (any error with the code 10060 changes the RPC status to "Unable to Connect")

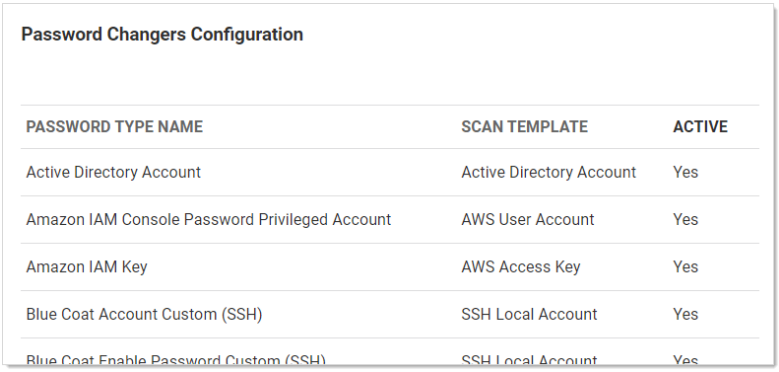
Procedure

To configure the unknown errors to trigger connection failures:

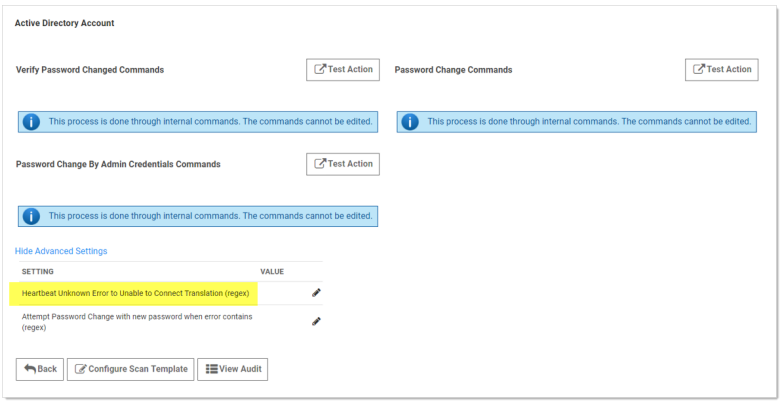
1. Go to **Admin > Remote Password Changing**. The Remote Password Changing Configuration page appears:



2. Click the **Configure Password Changers** button. The Password Changers Configuration page appears:



3. Click the link for the desired password type. Its Account page appears:



4. If necessary, click the **Advanced Settings** link.
5. Click the pencil icon next to **Heartbeat Unknown Error to Unable to Connect Translation (regex)**. The Value text box appears.
6. Determine the desired text string to search for.
7. Type the desired regex in the **Value** text box.
8. Click the **Save** icon next to the text box

SSH Key Rotation

SSH Key Rotation allows you to manage your Unix account private keys and passphrases, as well as the passwords for the associated accounts. With key rotation, whenever the password is changed on the secret (manually, during a scheduled auto-change, or when checking in a secret that changes the password on check-in), the public/private key pair will be regenerated and the private key encrypted using a new passphrase. The public key will then be updated on the Unix machine referenced in the secret.

There are two topics addressed here:

- "Basic SSH Key Rotation" below: A step-by-step tutorial on quickly getting started with the default SSH key rotation secret types and password changers.
- "Custom SSH Key Rotation" on page 1053: Provides additional information for users who need to customize the default commands for their environment.

Basic SSH Key Rotation

This topic is a tutorial on how to quickly get started using SSH key rotation to change a Unix account's public and private key and automatically update a remote machine using Remote Password Changing (RPC).

Introduction

SSH key rotation allows you to manage your Unix account private keys and passphrases as well as their passwords. With key rotation, whenever the password is changed on the secret (manually, during a scheduled auto-change, or when checking in a secret that changes the password on check-in), the public/private key pair is regenerated and the private key encrypted using a new passphrase. The public key is then updated on the Unix machine referenced on the secret.

This document is a tutorial showing you how to quickly get started using SSH key rotation using our default key rotation password changers. For an in-depth description of SSH key rotation including modifying the command sets for your environment, see our "Custom SSH Key Rotation" on page 1053 topic.

Requirements

To use our default SSH key rotation commands, the following minimum requirements must be met on the machine being managed:

- SSH key logins should be enabled on the target using keys in OpenSSH format. While a secret can initially be created with keys in PuTTY format, these keys will be converted to OpenSSH format during rotation. Despite this conversion, the PuTTY launcher will continue to function properly, ensuring compatibility and uninterrupted access.
- Public keys should be stored in `[~userhome]/.ssh/authorized_keys` (not `authorized_keys2`).
- Grep and Sed should be installed on the target.
- If doing a privileged SSH key rotation, where a privileged user sets the key for another user, the privileged user must have sudo permissions that do not prompt for a password and the permissions to edit the user's `authorized_keys` file with sudo.

If a system does not meet these requirements it may still be possible to do key rotation by modifying the key rotation command sets. Our "Custom SSH Key Rotation" on page 1053 topic describes how to do this.

Configuring a Secret for SSH Key Rotation

Secret Server comes with two secret templates for SSH key rotation: **Unix Account (SSH key rotation)** and **Unix Account (Privileged Account SSH Key Rotation)**.

Use **Unix Account (SSH Key Rotation)** if:

- The account is able to change its own password and modify its own `authorized_keys` file.
- The account password and key should only be changed by Secret Server (Secret Server will always have the current password and keys).

Use **Unix Account (Privileged Account SSH Key Rotation)** if:

- The account is not able to change its own password or modify its own `authorized_keys` file.
- The account password and key may be changed outside of Secret Server, and Secret Server may not have the current account credentials. A privileged account that is able to change the password and `authorized_keys` files of other users will still be able to change the account credentials.

SSH Key Rotation Using the Secret's Credentials

Creating the Secret

1. Create a new secret in Secret Server using the **Unix Account (SSH Key Rotation)** template.
2. Enter the account user name and password.
3. Upload the private key file.

4. If the private key is encrypted using a passphrase, enter the passphrase.
5. Uploading a public key is optional but recommended. If a public key is not provided, Secret Server will regenerate it from the private key during key rotation, but if the key in `authorized_keys` is not in the same format as the generated key or does not match exactly (including comments), the rotation will fail because it could not find the public key that needs to be removed.
6. After the secret is created you should see a successful heartbeat status. If heartbeat isn't running, make sure heartbeat and RPC are enabled under **Admin > Remote Password Changing**.

Rotating the Key

1. Go to the **Remote Password Changing** tab and click **Change Password Remotely**.
2. Enter the new password or click **Generate** next to the **Next Password** field to generate a random password.
3. Click to select **Generate New SSH Key** to create a new, random SSH key. If you want to supply your own private key, uncheck this option and paste the key into the **Next Private Key** text box that appears.
4. If you have unchecked **Generate New SSH Key** you must enter the passphrase that was used to encrypt the private key at the time it was created. Leave this field blank if the private key was not encrypted with a passphrase. If you have checked **Generate New SSH Key** you have the option to enter your own passphrase, leave it blank (for an unencrypted private key), or click the **Generate** button next to the field to create a new, random passphrase. If you want to change the key without changing the passphrase, you must put the current passphrase in the **Next Private Key Passphrase** text box.
5. Click **Change** to start the key rotation and a password change. After you start the change, you can check the status either in **Admin > Remote Password Changing** or on the **Remote Password Changing** tab of the secret.
6. Once the password change / key rotation is complete the heartbeat status should be successful. You can check the audit log to see notes that the key was rotated and start a session using the key with the PuTTY Launcher.

SSH Key Rotation Using a Privileged Account

To use **Unix Account (Privileged Account SSH Key Rotation)**, you must have a secret that is able to use the sudo command to access other accounts' `authorized_keys` files and change their passwords. This can be any type of Unix secret and can use a password and/or private key to authenticate. If you have a secret that meets these requirements, you can set up SSH key rotation using a privileged account as follows.

Creating the Secret

1. Create a new secret in Secret Server using the **Unix Account (Privileged Account SSH key rotation)** template.
2. Enter the account user name and password.
3. Upload the private key file.
4. If the private key is encrypted using a passphrase, enter the passphrase.
5. Uploading a public key is optional, but recommended. If it is not provided, Secret Server will regenerate it from the private key during key rotation, but if the key in `authorized_keys` is not in the same format as the generated key or does not match exactly (including comments), the rotation will fail because it could not find the public key that needs to be removed.

6. After the Secret is created you should see a successful heartbeat status. If heartbeat is not running, make sure that heartbeat and RPC are enabled under **Admin > Remote Password Changing**.
7. Next go to the **Remote Password Changing** tab and choose the privileged secret that can authenticate to the machine and modify the user's `authorized_keys` file.
8. Click the **Back** button after adding the associated secret.

Rotating the Key

1. Go to the **Remote Password Changing** tab and click **Change Password Remotely**.
2. Enter the new password or click **Generate** next to the **Next Password** field to generate a random password.
3. Click to select **Generate New SSH Key** to create a new, random SSH key. If you want to supply your own private key, uncheck this option and paste the key into the **Next Private Key** text box that appears.
4. If you have unchecked **Generate New SSH Key** you must enter the passphrase that was used to encrypt the private key at the time it was created. Leave this text box blank if the private key was not encrypted with a passphrase. If you have checked **Generate New SSH Key** you have the option to enter your own passphrase, leave it blank (for an unencrypted private key), or click the **Generate** button next to the field to create a new, random passphrase. If you want to change the key without changing the passphrase, you must put the current passphrase in the **Next Private Key Passphrase** text box.
5. Click **Change** to start the key rotation and a password change. After you start the change, you can check the status either in **Admin > Remote Password Changing** or on the **Remote Password Changing** tab of the secret.
6. Once the password change / key rotation is complete the heartbeat status should be successful. You can check the audit log to see notes that the key was rotated and start a session using the key with the PuTTY Launcher.

Troubleshooting

- The SSH Password Changers are targeted to OpenSSH. If using a different SSH library or if the user keys are not in the user's `/.ssh/authorized_keys` file you can check the commands used and modify them as appropriate under **Admin > Remote Password Changing** and clicking **Configure Password Changers**. The password changers used are **SSH Key Rotation** and **SSH Key Rotation Privileged Account**.
- Errors are logged to **Admin > Remote Password Changing**. Additional logs can be found in the Secret Server directory in the log subfolder. That is: `C:\inetpub\wwwroot\secretserver\log`.
- A change was made to how SSH script variables are named to differentiate them from tokens when testing command sets on the Configure Password Changers page. Non-token script variables should begin with an underscore. Anything in the script beginning with a dollar sign not followed by an underscore will be treated as a token and displayed as a field in the test dialog. For example:
  - `$USERNAME`References the username from the secret.
  - `$[1]$USERNAME` References the username from the first linked secret.
  - `$_USERNAME` References a bash variable defined in the script.
- The default command set for the SSH key rotation privileged account password changer assumes that the sudo command will not prompt for a password. If your environment prompts for a password when using sudo the command sets will need to be modified to supply the password. If your environment caches the sudo credentials, the easiest way to handle this is to add the following two lines at the top of each command set on

the SSH key rotation privileged account password changer:

```
sudo -i echo
$[1]$PASSWORD
```

This caches the credentials for the rest of the script.

Custom SSH Key Rotation

This topic discusses how to change public keys for Unix accounts using Remote Password Changing (RPC) in Secret Server. For a step-by-step tutorial on quickly getting started with the default SSH key rotation secret types and password changers, see our "Basic SSH Key Rotation" on page 1049. The current topic provides additional information for users who need to customize the default commands for their environment.

Introduction

SSH key rotation allows you to manage your Unix account private keys and passphrases as well as their passwords. With key rotation, whenever the password is changed on the secret (manually, during a scheduled auto-change, or when checking in a secret that changes the password on check-in), the public/private key pair is regenerated and the private key encrypted using a new passphrase. The public key will then be updated on the Unix machine referenced on the secret.

Secret Server provides secret templates and password changers for SSH key rotation.

Requirements

To use our default SSH key rotation commands, the following minimum requirements must be met on the machine being managed:

- SSH key logins should be enabled on the target using keys in OpenSSH format. A secret can be created with keys in PuTTY format but they will be converted to OpenSSH when the key is rotated.
- Public keys should be stored in `[~userhome]/.ssh/authorized_keys` (not `authorized_keys2`).
- Grep and Sed should be installed on the target.
- If doing a privileged SSH key rotation, where a privileged user sets the key for another user, the privileged user must have sudo permissions that do not prompt for a password and the permissions to edit the user's `authorized_keys` file with sudo.

The default command sets have been tested in the following Linux environments:

- CentOS Linux release 7.0.1406
- FreeBSD bsdRadiusServer 9.3-RELEASE-p5 i386
- Linux Ubuntu 3.13.0-32-generic

If a system does not meet these requirements or has a different configuration than the tested Linux environments, it may still be possible to do key rotation by modifying the key rotation command sets. The command sets that may need to be edited and the process for doing so are described later in this topic.

Secret Templates

Secret Server includes two secret templates for SSH key rotation: **Unix Account (SSH Key Rotation)** and **Unix Account (Privileged Account SSH Key Rotation)**. The first template changes the password and key on the account using the account's credentials. Use this template if both of the following conditions apply:

- The account is able to change its own password and modify its own `authorized_keys` file.
- The account password and key should only be changed by Secret Server, which will always have the current password and keys.

Unix Account (Privileged Account SSH Key Rotation) uses an additional secret to provide the credentials for the connection that performs the password change and key rotation commands. You should use this template if either of the following conditions apply:

- The account is not able to change its own password or modify its own `authorized_keys` file.
- The account password and key may be changed outside of Secret Server, and Secret Server may not have the current account credentials. A privileged account that is able to change the password and `authorized_keys` files of other users will still be able to change the account credentials.

Creating a New SSH Key Rotation Secret

When creating a new secret based on either of these templates you will see the following form:

1. Type the secret name in the **Secret Name** text box.
2. Type the machine in the **Machine** text box.
3. Type the username in the **Username** text box.
4. Click the **Generate** button to create a user password.
5. Click **Private Key** link to upload a file containing the private key.
6. If you are creating a new secret and want to generate a new, random private key, click to select the **Generate New SSH Key** check box. This disables the "Change" links for both the private and public keys.
7. If your uploaded private key was encrypted with a passphrase or you are generating a new key and wish to encrypt it with a passphrase, type that passphrase in the **Private Key Passphrase** text box. Otherwise, leave it empty.
8. If you are creating a new key and want to create a random passphrase for it, click this **Generate** button.
9. If you are uploading a private key, click the **Public Key Change** link to upload the corresponding public key. Uploading a public key is optional, but recommended. If not provided, Secret Server regenerates it from the private key during key rotation, but if the key in the `authorized_keys` file is not in the same format as the generated key, the old key will not be removed when the new key is added.

If neither private key nor public key is attached to the secret, a key rotation creates a new key pair, attaches them to the secret, and adds the new public key to `authorized_keys`.

After the secret is created, you should see a successful heartbeat status. If heartbeat is not running, make sure that heartbeat and RPC are enabled under **Admin > Remote Password Changing**.

> Heartbeat status when either the private key/passphrase or password are incorrect is indeterminate and based on the host configuration. If the system allows log on as long as one of the two is correct, it will return a successful heartbeat when the password is wrong but the key is valid and vice-versa.

If you are adding a secret using the **Unix Account (Privileged Account SSH Key Rotation)** you will also need to specify which privileged account to use during key rotation. To do this:

1. Switch to the **Remote Password Changing** tab.
2. Click the **Edit** button.
3. Click the **No selected secret** link.
4. Choose a privileged Secret that can authenticate to the machine and use the sudo command to access other accounts' authorized_keys files and change their passwords. This can be any type of Unix secret and can use a password and/or private key to authenticate.
5. Click the **Back** button to exit edit mode.

Editing the SSH Key Rotation Templates

To edit a template, go to **Admin > Secret Templates**, choose the template you want to edit in the dropdown list of templates, then click **Edit**.

You can add, remove, or edit any fields you like, but if you change or replace any of the following fields you will need to update the password changer mapping for the template:

- Machine
- Username
- Password
- Private Key
- Private Key Passphrase
- Public Key

> Private key and public key must remain field type "File".

If you change any of the fields listed above, click the **Configure Password Changing** button to map the fields to the password changer. Click the **Edit** button and assign all the password changer fields to the corresponding fields on the secret template.

Password Changers

Secret Server includes two password changers for SSH key rotation: **SSH Key Rotation** and **SSH Key Rotation Privileged Account**. The **Unix Account (SSH Key Rotation)** secret template uses the **SSH Key Rotation** password changer and the **Unix Account (Privileged Account SSH Key Rotation)** secret template uses the **SSH Key Rotation Privileged Account** password changer. Each of these password changers includes a set of command sets designed to change the password and public key on an account using the secret's credentials and using sudo with a privileged account, respectively.

Although you can edit these password changers through **Admin > Remote Password Changing > Configure Password Changers**, clicking on the password changer and then **Edit** and **Edit Commands**, the recommended practice is to copy the existing password changers and then modify the copies. To do this:

1. Go to **Admin > Remote Password Changing**.
2. Click the **Configure Password Changers** button.
3. Scroll down to the bottom of the page and click the **New** button.
4. Select the password changer you want to copy in the **Base Password Changer** list and give the new password changer a name.
5. Click **Save**. This creates a new password changer and copies the command sets from the original password changer.
6. Enter the authentication information (see below for more information), clicking the **Save** button in each authentication field set.
7. Make the required changes to the command sets for your environment by editing existing lines, deleting lines, adding new lines, and rearranging lines.

Authentication

The authentication section defines the credentials that will be used to connect to the machine and run the command set. These can be either the credentials of the secret or credentials from an associated secret. The command sets on the **SSH Key Rotation** password changer are designed to be run with the secret's credentials. Any customized password changer based on this password changer should use the secret's credentials in the authentication section.

For more information about tokens beginning with a dollar sign used in the above screenshot, see the "Dependency Token List" on page 1490.

The command sets on the **SSH Key Rotation Privileged Account** password changer are designed to be run with the credentials of an account that can change the password and public key on behalf of other users. Any customized password changer based on **SSH Key Rotation Privileged Account** should use the credentials off one of the secret's associated secrets. This is typically the first associated secret but can be any associated secret if your password changer requires more than one associated secret. The exception to this is validation which does not run a command set by default and uses the secret's credentials. If you modify validation to use a command set you will need to change the default authentication for validation if the command set uses sudo.

Here is a typical authentication for **SSH Key Rotation Privileged Account** (except validation, which is identical to the authentication block used by **SSH Key Rotation**):

Username: `$[1]$USERNAME`

Password: `$[1]$PASSWORD`

Key: `$[1]$PRIVATE KEY`

Passphrase: `$[1]$PRIVATE KEY PASSPHRASE`

Command Sets

Overview

To handle cleanup and error-handling for key rotation two command set types were added: **Post Successful Change** and **Post Fail Change**. A password change using one of the two new password types will execute as follows:

1. Connect to the box using the specified credentials (secret credentials or an alternate secret's credentials).
2. Run the Password Change command set. This will add the new public key to the authorized_keys file. It does not change the password yet nor does it remove the old public key.
3. Verify using the new public key and old password. This verifies that the new key was added correctly.
4. If the verify check is successful, run the Post Success Change command set. This changes the password and, if successful, removes the old public key from authorized_keys if present.
5. If the verify check is unsuccessful, run the Post Failure Change command set. This removes the new public key from authorized_keys and does not change the password.
6. Return the success or failure of the overall process.

Post Successful Change and **Post Failure Change** are advanced command sets that are hidden by default. To see them, scroll to the bottom of the page and click the **Advanced Post Change Commands** link.

Password Change Command Set

In other password types the Password Change command set is only responsible for changing the password. For the two key rotation password changers, the default Password Change command set checks for the existing public key on the secret, and if found, will append a new public key to authorized_keys (the old key is then removed in the Post-Reset Command Set). The password change is done in the Post Successful Change command set only after the key change has been validated.

Verify Password Changed (Heartbeat) Command Set

Following this Reset Command Set, a Verify Password Changed is performed by attempting to connect to the host using the credentials on a secret to validate the new public key that was added to authorized_keys. If a command set is present, those commands are then run after connecting and the validation is a success only if both the connection and the command set are successful. (The command set is normally used when a secret uses alternate secrets as credentials in which case the alternate credentials are specified for authentication and command set does the actual validation of the secret.)

In the case of SSH key rotation, this validation heartbeat is run immediately after the reset command set using the current username, current password, new private key and new passphrase to connect for validation. If connection is successful, the validation is considered successful and Post Successful Change command set is run next to remove the old public key (if current private/public keys exist on the secret) from authorized_keys). If validation is not successful, the Post Fail Change command set is run to remove the new public key added during the reset.

Post Successful Change Command Set

If the Password Change command set and Verify Password Changed are both successful, the Post Successful Change command set is run. This command set finalizes the key rotation by changing the password on the account

and removing the old public key from authorized_keys.

Post Fail Change Command Set

If the Password Change command set is successful but the Verify Password Changed fails, the Post Fail Change command set is run. This command set rolls back the changes made in the Password Change command set by removing the new public key from authorized_keys.

For more information about customizing SSH command sets, see "Creating a Custom SSH Password Changer" on page 1032.

Notes

SSH Key Rotation scripts will typically be more complex than password change command sets that do not do key rotation. These scripts will often include tokens representing values from the secret and associated secrets as well as commands to verify success or failure of previous commands using `$$CHECKFOR` and `$$CHECKCONTAINS`. For more information about these features see the **Editing a Custom Command** topic.

The default command sets for **SSH Key Rotation Privileged Account** use sudo to execute several commands. These command sets assume that the sudo command will not prompt for a password. If your environment prompts for a password when using sudo the command sets will need to be modified to supply the password. If your environment caches the sudo credentials, the easiest way to handle this is to add the following two lines at the top of each command set on the SSH key rotation Privileged Account password changer:

```
sudo -i echo
$[1]$PASSWORD
```

This will pass the credentials from the first associated secret when prompted by sudo and cache the credentials for the rest of the script.

Troubleshooting

- The SSH Password Changers are targeted to OpenSSH. If using a different SSH library or if the user keys are not in the users `/.ssh/authorized_keys` file you can check the commands used and modify them as appropriate under **Admin > Remote Password Changing** and clicking **Configure Password Changers**. The password changers used are **SSH Key Rotation** and **SSH Key Rotation Privileged Account**.
- Errors are logged to **Admin > Remote Password Changing**. Additional logs can be found in the Secret Server directory in the log subfolder. For example: `C:\inetpub\wwwroot\secretserver\log`.
- A change was made to how SSH script variables are named in order to differentiate them from tokens when testing command sets on the Configure Password Changers page. Non-token script variables should begin with an underscore. Anything in the script beginning with a dollar sign not followed by an underscore will be treated as a token and displayed as a field in the test dialog. For example:
  - `$USERNAME` – References the username from the Secret.
  - `$[1]$USERNAME` – References the username from the first linked Secret.
  - `$_USERNAME` – References a bash variable defined in the script.

Secrets

Secrets in Secret Server are individually named sets of sensitive information, such as passwords, API keys, SSH keys, and other authentication credentials. These secrets are created using secret templates, which define the fields, launchers, and remote password changers for different types of secrets. Secret Server ensures that all secret data is securely encrypted before being stored in the database, and it maintains a detailed audit trail for access and history. Secrets can be centrally managed through sharing settings and folder structures, allowing for inheritance of permissions from parent folders. This robust management system helps organizations securely store, manage, and control access to privileged credentials, reducing the risk of data breaches and ensuring compliance with security policies.

Folders

Folders in Secret Server allow you to organize your secrets into logical groups and control access through permissions assigned to the folders. Secret folders allow you to create containers of secrets based on your needs. They help organize your customers, computers, regions, and branch offices, to name a few. Folders can be nested within other folders to create sub-categories for each set of classifications.

You can assign secrets to these folders and sub-folders. Folders allow you to customize permissions at the folder level, and all secrets within can inherit the folder's permissions. Setting permissions at the folder level ensures future secrets placed in that folder have the same permissions, simplifying management across users and groups.Thoughtfully organizing your secrets into folders and setting granular permissions helps ensure the right people have access to the credentials they need while maintaining security.

You can "favorite" a folder in the main menu by right clicking it.

Folder Permissions

If the new folder is a subfolder, it can use the sharing settings of its parent folder if you enable the **inherit permissions from parent** setting for the folder.

Folders can apply one of the following permissions to users or groups in the folder's **Permissions** table:

You can access a folder's permissions table by accessing the folder, clicking on the three horizontal dots by its name and selecting **Edit Folder** from the dropdown options.

- **View**: Allows the user to see the folder and the secrets in that folder which inherit its permissions. Users need to have this permission for the parent folder to be able to see any subfolders available. Permissions granted to the root/parent folder will be inherited by subfolders.
- **Edit**: Allows the user to create new folders in the root/parent folder, which forces the **Inherit Permissions from Parent** setting on the new folder. This permission also allows for creating new and moving secrets into that folder, as well as renaming the folder.
- **Add Secret**: Allows the user to add a secret into a folder, but does **NOT** grant access to the added secret.

- **Owner**: Allows the user to create new folders in the root folder without forcing inheritance. It also allows the user to move, delete, or rename the folder, as well as change the permissions and inheritance settings on the folder.

Depending on your configuration, these settings could affect the permissions of subfolders and secrets contained in the root folder. Folders are not visible to users that do not have at least the **View** permission. This allows users to create and manage their own folders without making them visible to all users. Some folder permissions include other permissions.

Table: Included Folder Permissions (Ordered from lowest to highest permissions required)

Permission	Description	Included Permissions
Add Secret	Allows adding new secrets	View
List (Secret)	Allows viewing secret names	None
View	Allows viewing the folder	None
View (Secret)	Allows access to secret contents	List
Edit	Allows for editing secrets, creating and/or renaming subfolders	Add Secret, View
Edit (Secret)	Allows editing of a secret	List, View
Owner	Allows full control over the folder	Add Secret, Edit, View
Owner (Secret)	Allows full control over the secret	Edit, List, View

Personal Folders

In Secret Server, a *personal folder* is a folder that one (and only one) individual has owner access to. No other user can modify sharing permissions on these folders. Users can add subfolders to their personal folder. The purpose of this folder is to allow a user to securely store work-related secrets that other users do not require access to.

If in *break-the-glass* mode, an unlimited admin can access a user's personal folder in order to recover secrets if needed.

Required Role Permissions for Managing Folders

Folder management is subject to these role permissions:

- The **Administer Folders** role permission allows a user to create new folders and manage folders but specific folder permissions still apply.
- Any user with the Administer Folders role permission can create new folders, but to create folders at the root level, the user also needs the **Create Root Folders** permission.
- Any user who has the **Create Root Folders** permission can add new folders to any folders where they have **Edit** or **Owner** permissions.
- Users must have the **Owner** permission to delete a folder.

- Users can also move folders if they have the **Owner** permission on the source folder and the **Edit** or **Owner** permission on the target folder (where they are moving it). The folder automatically inherits permissions from its parent when it is moved, which is the same as when secrets are moved.

Managing Folders

Managing folders in Secret Server allows you to organize secrets into logical groups and control access through permissions assigned at the folder level. Folders can be nested to create sub-categories, helping to structure secrets based on various criteria such as departments, projects, or regions. Permissions can be set to allow users to view, edit, add secrets, or have full ownership of the folder, ensuring that only authorized personnel have access to sensitive information. Additionally, folders can inherit permissions from their parent folders, simplifying the management of access controls. This hierarchical organization and permission structure help maintain security and streamline the management of secrets across the organization.

Assigning Secret Policies to the Secrets in the Folders

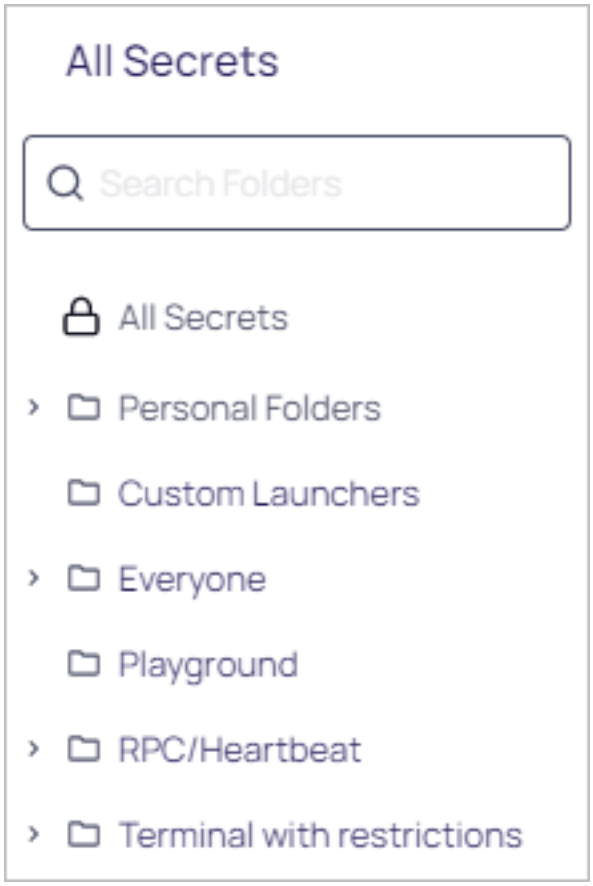
1. Open the folder containing the secrets you would like to assign a policy to.
2. Click on the ellipsis button next to the folder name and select Edit Folder. The Folder Details page appears:
3. Click the **Secret Policy Edit** link. The Edit Folder popup appears:
4. Click to select the desired policy in the **Secret Policy** dropdown list.
5. Click the **Save** button.

Creating Folders

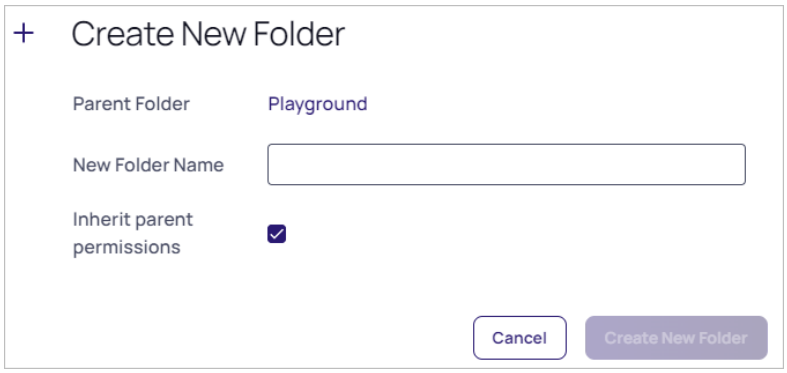
To create a folder:

To create folders, you must have a role with the "administer folder" permission. You also must have edit or owner permission for the parent folder.

1. Click **>>** on the **Secrets** menu item to view the Secret Folder-Tree Panel.



2. Drill down and click the folder you wish to add a subfolder to. We chose the Playground folder.
3. Right click the folder and select **Add Subfolder**. The Create New Folder popup appears:



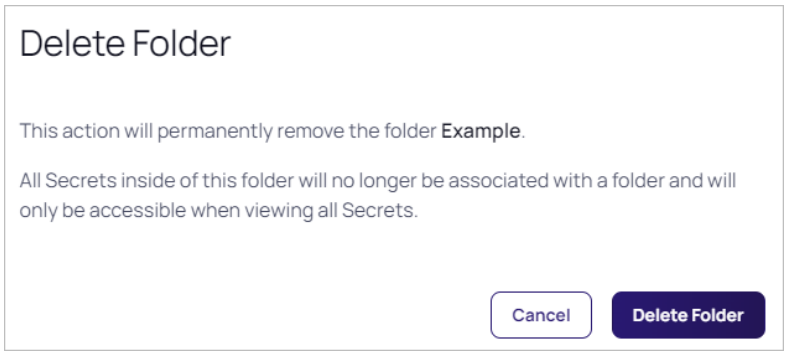
4. Type the new folder's name in the **New Folder Name** text box.
5. If desired, click to toggle the **Inherit Parent Permissions** check box.
6. Proceed to "Editing Folder Permissions" on the next page to customize permissions for the new folder.

Deleting Folders

To delete a folder:

> To delete folders, you must have a role with the "administer folder" permission. You also must have edit or owner permission for the parent folder.

1. Navigate to the folder in the folder tree on the main menu.
2. Right click the folder and select **Delete Folder**. The Delete Folder pop-up page appears:



3. Click the **Delete Folder** button.

Editing Folder Permissions

> To edit folder permissions, you must have a role that has the **Administer Folder** permission, and the **Owner** permission for the folder. If **Inherit Permissions from Parent Folder** is disabled, you must also have the Owner permission for the parent folder.

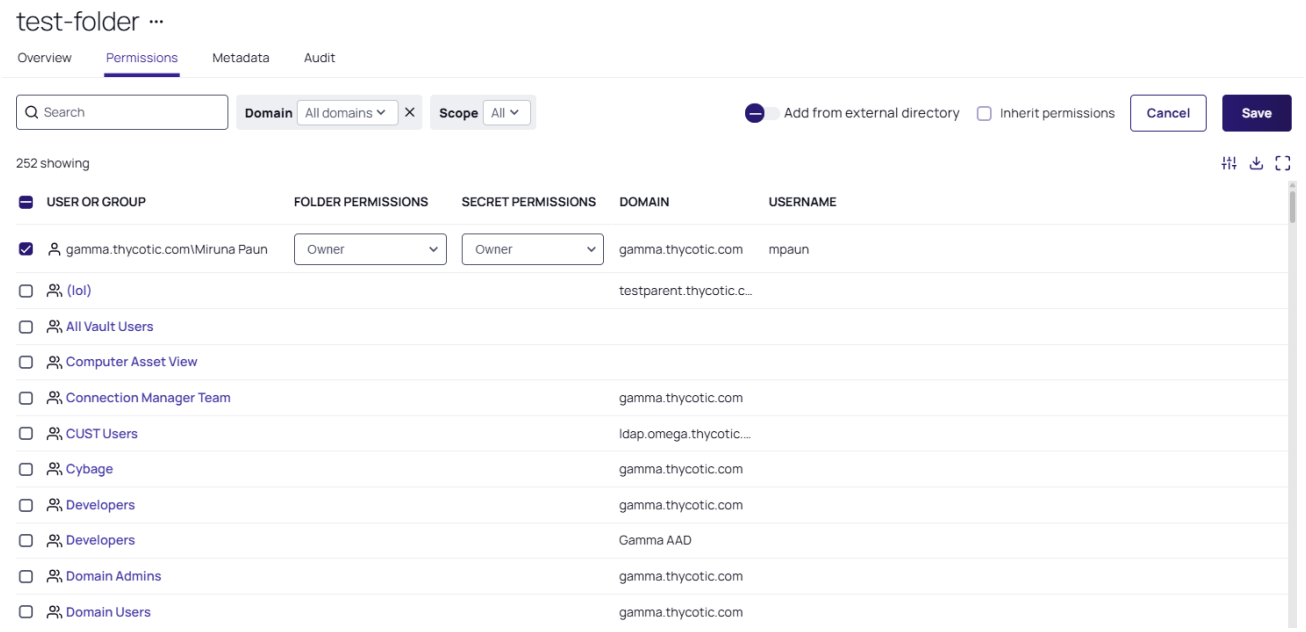
To edit folder permissions:

1. Navigate to the folder you wish to edit: **Secrets > Folders > [Your Folder]**.
2. Select the 3 horizontal dot menu for the folder located next to its name and choose **Edit Folder**. The Overview tab page appears.
3. Click the **Permissions** tab.
4. Click the **Edit** button. The Folder Permissions section becomes editable.

> The **Inherit Permissions** checkbox indicates if the permissions are applied from the parent folder. When inheriting permissions from a parent folder, the permissions cannot be updated here but only in the parent/root folder.

> Root folders cannot inherit permissions.

5. If necessary, click to deselect the **Inherit Permissions** checkbox. The permissions section becomes editable:



> The Inherit Permissions box needs to be deselected to view Allowable Templates.

6. Change the **Scope** to **All**, to see all available users and groups.
   a. To remove a user, deselect the checkbox next to that user.
   b. To add a user or group, select the checkbox next to each user/group and choose from the Folder and Secret permissions dropdown lists that appear.

      > Applying different filters, such as searching for a user, will not remove any pending changes.

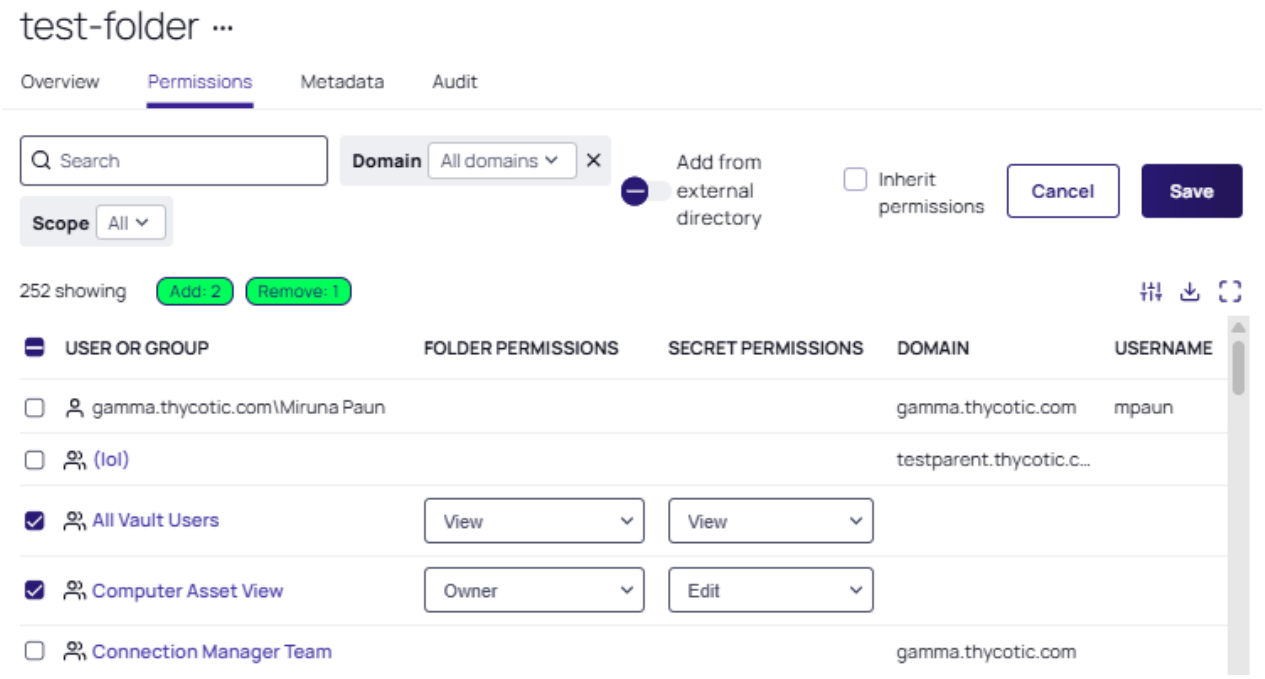
7. Select the permissions for each desired user/group:
   a. From the **Folder Permissions** dropdown list select one of the following:
      - **View** (folder)
      - **Add Secret** (to folder)
      - **Edit** (folder)
      - **Owner** (of the folder).

        These options are listed in order of increasing permissions.
   b. From the **Secret Permissions** dropdown list select one of the following:
      - **None**
      - **List** (secrets in folder)

- **View** (secrets in folder)
- **Edit** (secrets in folder)
- **Owner** (of secrets in folder).

  These options are listed in order of increasing permissions.

For example:



> The two rounded green rectangles shown in the image above, indicate the current uncommitted changes. Each user or group can have explicit folder and secret permissions. The secret permissions are applied to any secret within this folder that inherits permissions.

8. Click the **Save** button to commit your changes or the **Cancel** button to remove all pending changes.

Enabling Personal Folders

> The following cannot be edited on personal folders:
>
> - Secret policy assignment
> - Secret template restrictions
> - Folder permissions
>
> You can, however, edit some of them in the secrets themselves or on subfolders.

To use personal folders, you must first enable them:

1. Go to **Admin > Configuration > Folders Configuration > General Folder Settings**.
2. Click the **Edit** button in the **Folders Configuration** section. The entire section becomes editable.
3. Click to select the **Enable Personal Folders** check box.
4. (Optional) Type a new folder name in the **Personal Folder name** text box to customize the root-level folder that contains all personal folders.
5. (Optional) If you want to display a warning message to users when placing secrets in their personal folders:
   a. Click to select the **Should the warning message be displayed to the user when creating Secrets?** check box.
   b. (Optional) Edit the **Warning message text** box.
6. (Optional) Click the **Personal Folder Display Name** dropdown list to choose whether to display the personal folder display name or the username and domain.
7. Click the **Save** button. A personal folder for each user is now created in a root-level folder with the personal folder name specified.

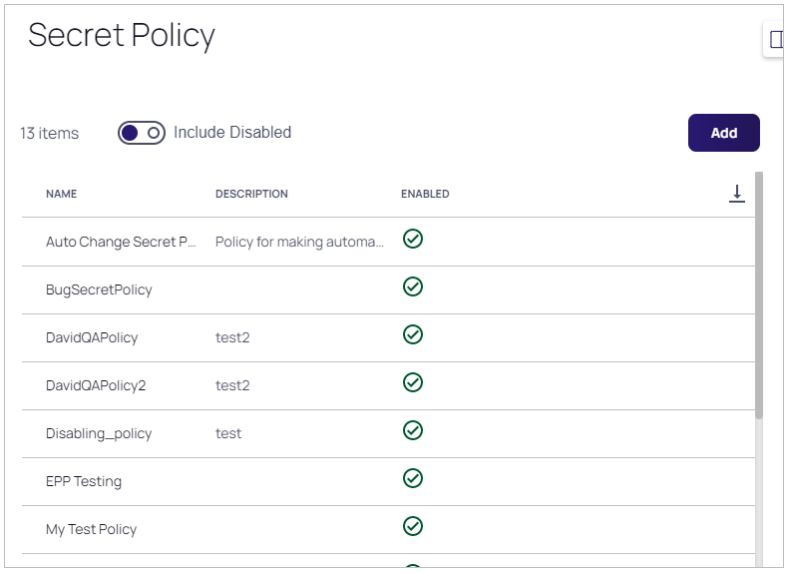
> When personal folders are enabled, a user requires the Personal Folders role permission in their role to be able to view and use their own personal folder.

Modifying Folders with Secret Policies

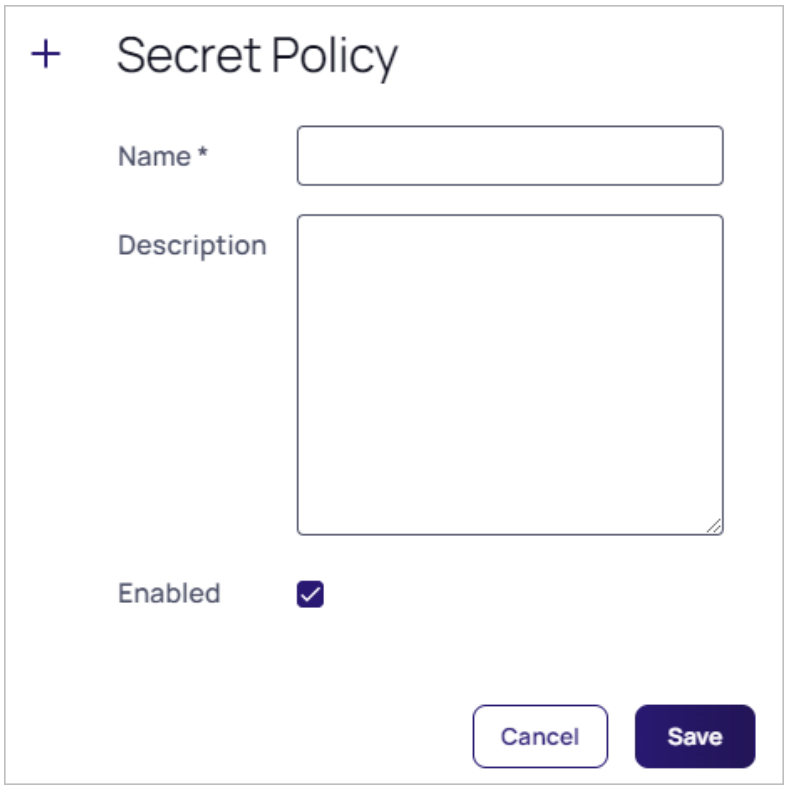
You can configure secret policies to apply RPC and security settings to an entire folder of secrets.

To create a new secret policy:

1. Click **Admin > Secret Policy**. A Secret Policy page appears:



2. Click the **Add** button. The (new) Secret Policy page appears:



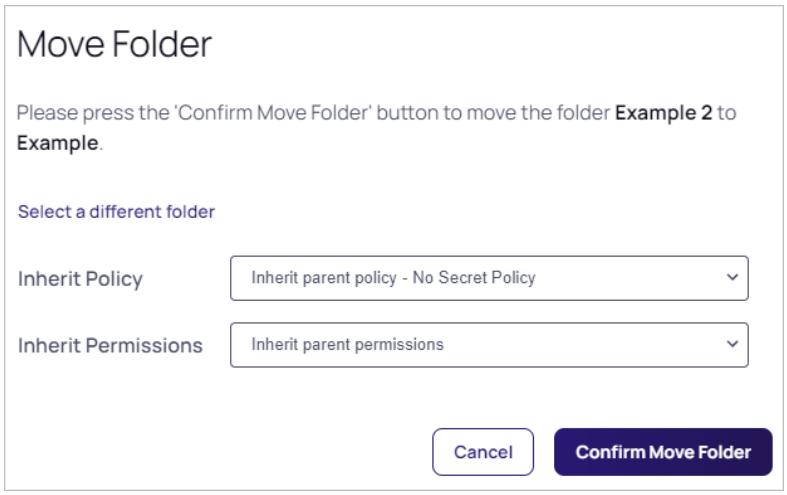
3. Type a name for the new secret policy in the **Name** text box.
4. (Optional) Type a description in the **Description** text box.
5. Ensure the **Enabled** check box is selected.

> To deactivate a policy that you no longer want, edit the policy and deselect the **Enabled** check box. For information about applying a secret policy to a folder, see "Editing Folder Permissions" on page 1063.

Moving Folders

To move a folder:

1. Ensure that you have edit permission for both the source and destination folders.
2. Expand the folder tree to make both folders visible.
3. Drag the source folder and drop it on the destination folder. The Move Folder page appears:

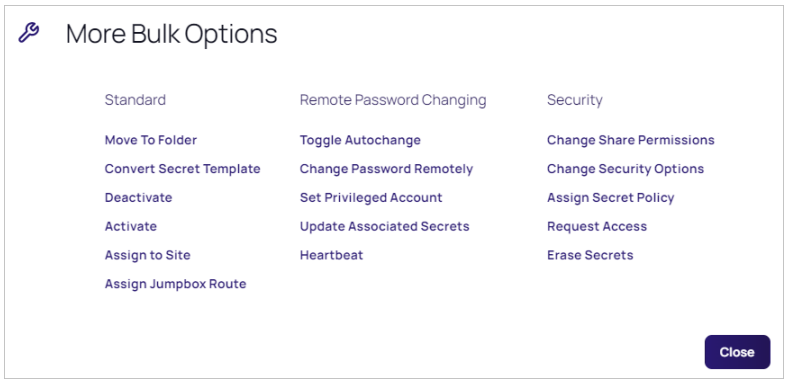


4. Click the **Inherit Policy** dropdown list to select how you want policy inheritance to work for the folder in the new location.
5. Click the **Inherit Permissions** dropdown list to select how you want permission inheritance to work for the folder in the new location.
6. Click the **Confirm Move Folder** button.

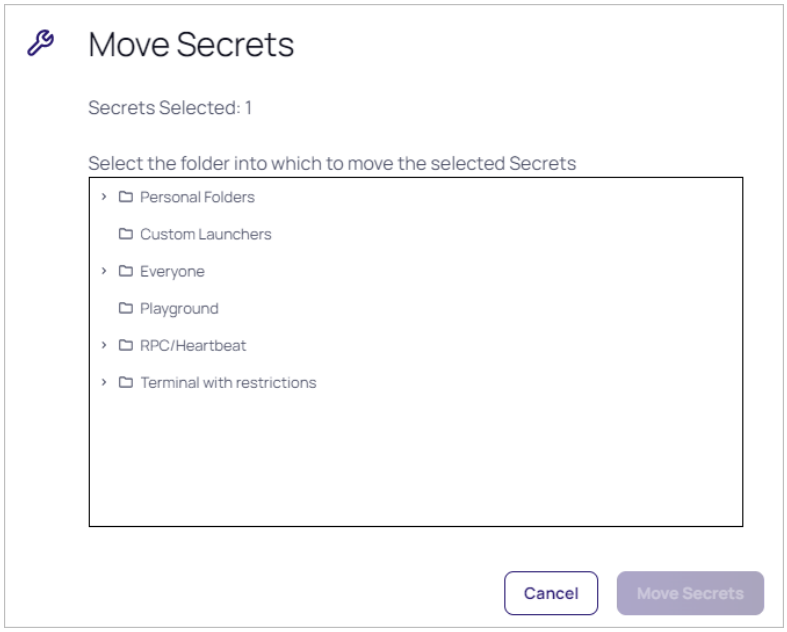
> You can also right click the source folder and select **Move** to see an alternative method of moving folders where you are prompted to select the destination folder.

Moving Secrets Between Folders

1. Consider the following before moving a secret between folders:
   - To add or move a secret to a folder, you must have edit permission on that folder (either direct or through inheritance).
   - To move a secret from a folder, you must have edit permission on that secret. If the secret has the "Inherit Permissions from folder" setting enabled, then you must have owner permission to move that secret to a new folder.
   - When a secret is moved to a folder, it automatically gets the "Inherit Permissions from folder" setting even if it had specific permissions before the move.
2. Navigate to the folder containing the secret or secrets you want to move.
3. Click to select the check box on the left for all the secret you want to move. The Bulk Actions button appears at the bottom of the page.
4. Click the **Bulk Actions** button. A popup appears:



5. Click the **Move to Folder** link. The Move Secrets pop-up page appears:



6. Navigate to and select the target folder for the secret or secrets.
7. Click the **Move Secrets** button. The Bulk Progress popup appears.
8. The secret moves to the selected folder.

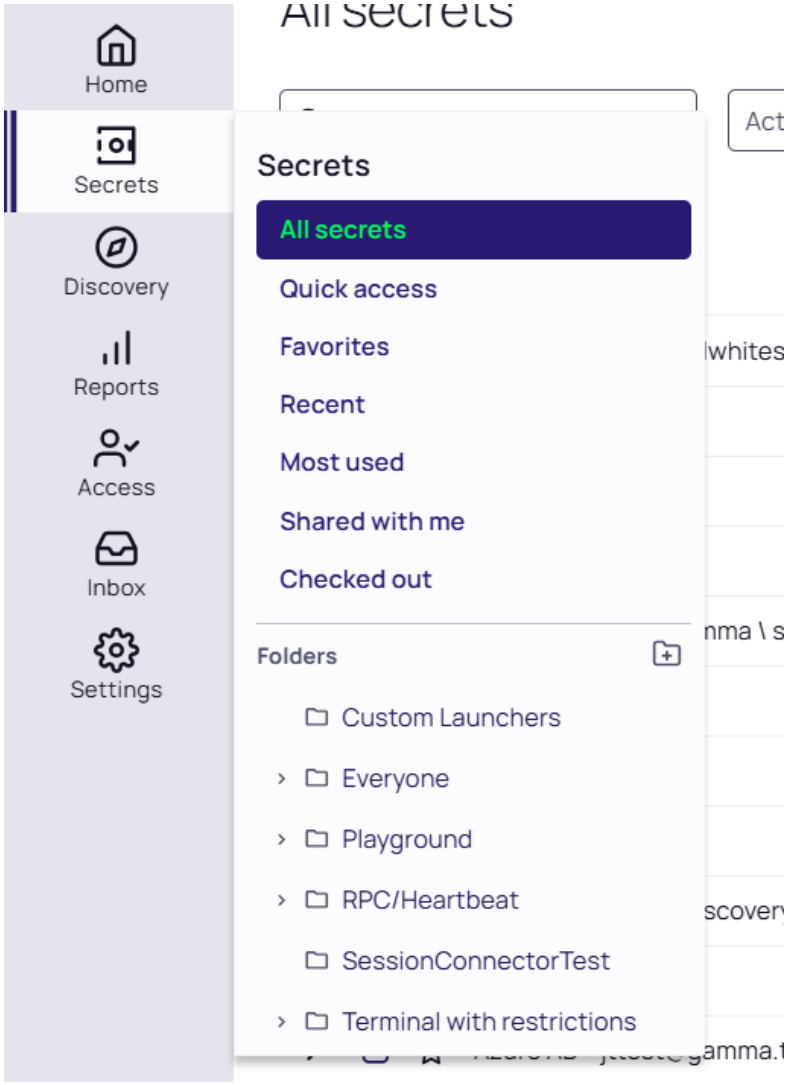
Pinning Folders

You can pin a folder, limiting the folder tree to show only the contents of that pinned folder. Applying filters to the table in one pinned folder will not affect others, and your settings are remembered when you return to the pinned folder.

Pinning Folders

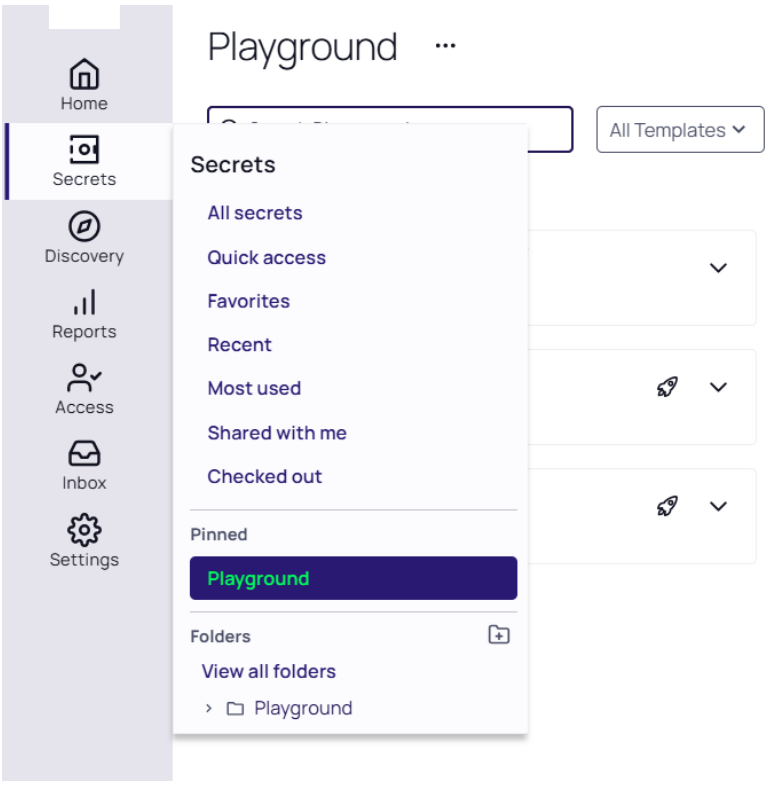
To pin a folder:

1. Hover over the **Secrets** menu item to view the Secret Folder-Tree Panel.



2. Drill down and click the folder you wish to pin. We chose the Playground folder.
3. Right click the folder and select **Pin folder**. The chosen folder will open.

   Now, when you hover over the **Secrets** menu item you can see that the selected **Playground** folder is in the **Pinned** section.



Unpinning Folders

To unpin a folder:

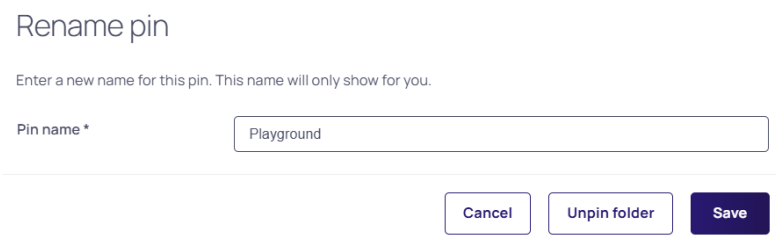
1. Hover over the **Secrets** menu item to view the Secret Folder-Tree Panel.
2. Select the pinned folder in the folders dropdown list, right click on it and select **Unpin folder** from the dropdown.

Alternatively, you can open the pinned folder, click the three dots next to the folder's name on the folder page, and select Unpin folder in the Rename Pin popup.

Custom Naming Pinned Folders

You do not have to use the default name of the top folder as the pin's name. To rename a pinned folder:

1. Hover over the **Secrets** menu item to view the Secret Folder-Tree Panel.
2. Select the pinned folder in the folders dropdown list. On the folder page, click on three dots next to the folder's name. The Rename Pin pop up will open.

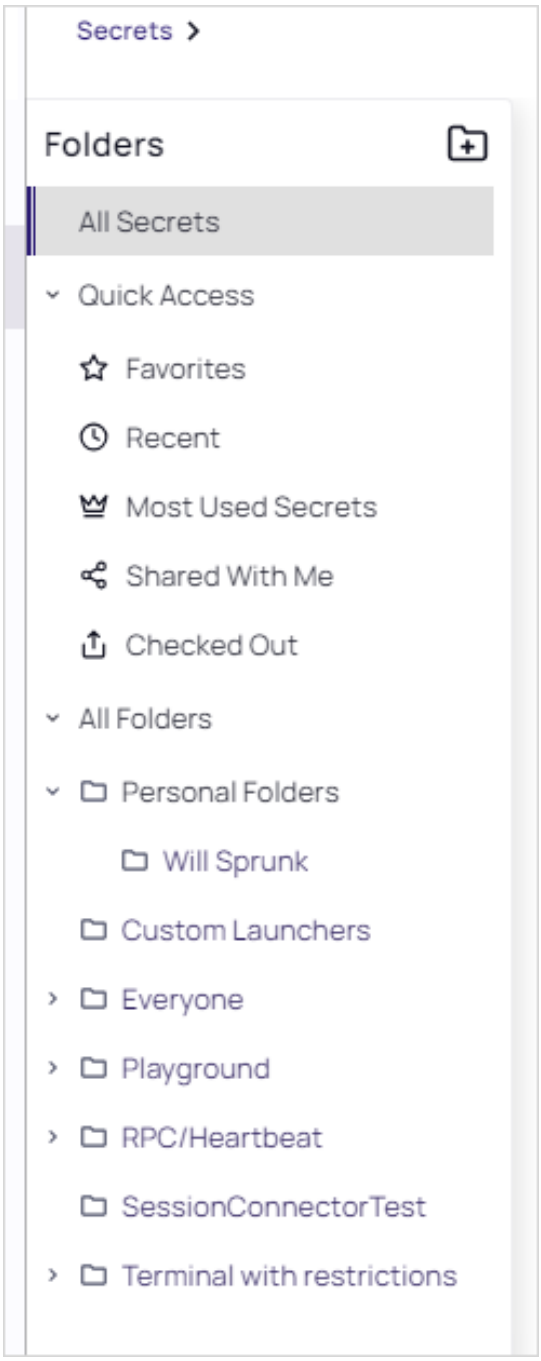


3. Type the new name in the **Pin Name** text box.
4. Click the **Save** button.

The folder dropdown list will now display the new name instead of the default top folder name for the pin.

Secret Folder-Tree Panel

The secrets menu item in the new UI has a collapsible secret folder-tree panel that occupies the full height of the page. This replaces the legacy folder tree at the bottom of the left navigation bar and delivers a significantly improved folder browsing experience. To view the panel, click the **>** icon on the bottom left of the page.



You can create folders, if you have permission to do so, by clicking the folder button on top of the folder panel.

Quick access provides a convenient view of high use secrets as well as keeping track of secrets that have been opened during the current session. Any pinned folder offers the same quick view but is scoped to the context of that folder. For example, only show me favorite or recent secrets from a single folder and its subfolders.

Direct links are available to common filtered views of secrets that allow for searching and advanced filtering within a specific context.

All folders to which you have access appear in the folder tree. A context menu offers options such as creating subfolders or pinning a specific folder. These options are also available when viewing a folder by clicking on the context menu next to the page title.

Secret Access and Workflow

Secret Server offers robust secret access and workflow features to enhance security and streamline the management of sensitive information. The platform allows administrators to configure access requests, requiring approval before users can access specific secrets. This can be set up to include multi-level and multi-user approval processes, ensuring that access is granted only after thorough validation. Additionally, Secret Server supports advanced workflows that can be customized to fit organizational policies, including features like timeouts and branching workflows to prevent bottlenecks. The system also integrates with ticketing systems such as ServiceNow and BMC, ensuring that access requests are tied to valid change or incident numbers. These workflows are fully audited, providing a comprehensive trail of all access requests, approvals, and denials, which is crucial for compliance and security.

Access Request Overview

The access request feature allows a secret to require approval prior to accessing the secret. Note the following:

- Establishing a workflow model, the user must request access from the approval group or groups.
- An email is sent to everyone in the approval groups, notifying them of the request.
- The request can be approved or denied by any members of the approval groups. Approvals can be set using the setting **Require Approval Type** to control who requires approval.
- Access is granted for a set time period.
- If **Owners and Approvers also Require Approval** is enabled, then even owners or those in an approval group needs to request access.
- Requestors cannot approve their own requests.

Approving Requests

Approving access requests in Secret Server involves a structured workflow to ensure secure and compliant access to sensitive information. To enable access requests, administrators can configure secrets to require approval before they can be accessed. When a user requests access, an email notification is sent to the approvers, who can then approve or deny the request. The approval process can be customized to include multiple steps and reviewers, ensuring that access is granted only after thorough validation. This workflow enhances security by maintaining an audit trail of all requests, approvals, and denials, and can be integrated with ticketing systems like ServiceNow or BMC for additional validation and compliance.

Duo Push Approvals

Users can now approve secret access requests and workflows using Duo push notifications. The push notification includes information, displayed on the user's screen, that helps the approver make the access decision.

Prerequisites

To use Duo push notifications:

- Duo must set up for Secret Server. See "Duo Security Authentication" on page 434.
- Duo user must be set up for Duo two-factor authentication. See "Duo Security Authentication" on page 434.
- The permission "Approve via DUO" must be granted to a role that is assigned to a group that includes all who will be approving requests via Duo. This allows enough flexibility so that those not wanting Duo push approvals can be configured to not receive them.

Assigning the Duo Approval Permission

To associate the permission with users:

1. Go to **Admin > Roles**.
2. Click the **Create New** button to create a new role. Name it "Duo Push Approver" or another name of your choosing.
3. Assign the **Approve Via DUO Push** permission to the new role.
4. Click the **Save** button.
5. If you choose to create a separate group for approvers, do this by navigating to **Admin > Groups**.
6. Click the **Create New** button to create a new group.
7. Add the desired users (chosen approvers) to that group.

   You can also assign users to the group later. This method is a shortcut when creating a group.
8. Click the **Save** button.
9. Go to **Admin > Roles**.
10. Click the **Assign Roles** button. The View Role Assignment page appears.
11. Click the **Role** dropdown list to select the role you created. Note that there are no groups or users.
12. Click the **Edit** button. The Role Assignment page appears.
13. Assign the **Approve via DUO Push** role to the **Assigned** list box.
14. Click the **Save Changes** button. Setup is now complete.

In addition to having the role you created, the user must be properly set up to receive Duo push notifications. See "Duo Security Authentication" on page 434.

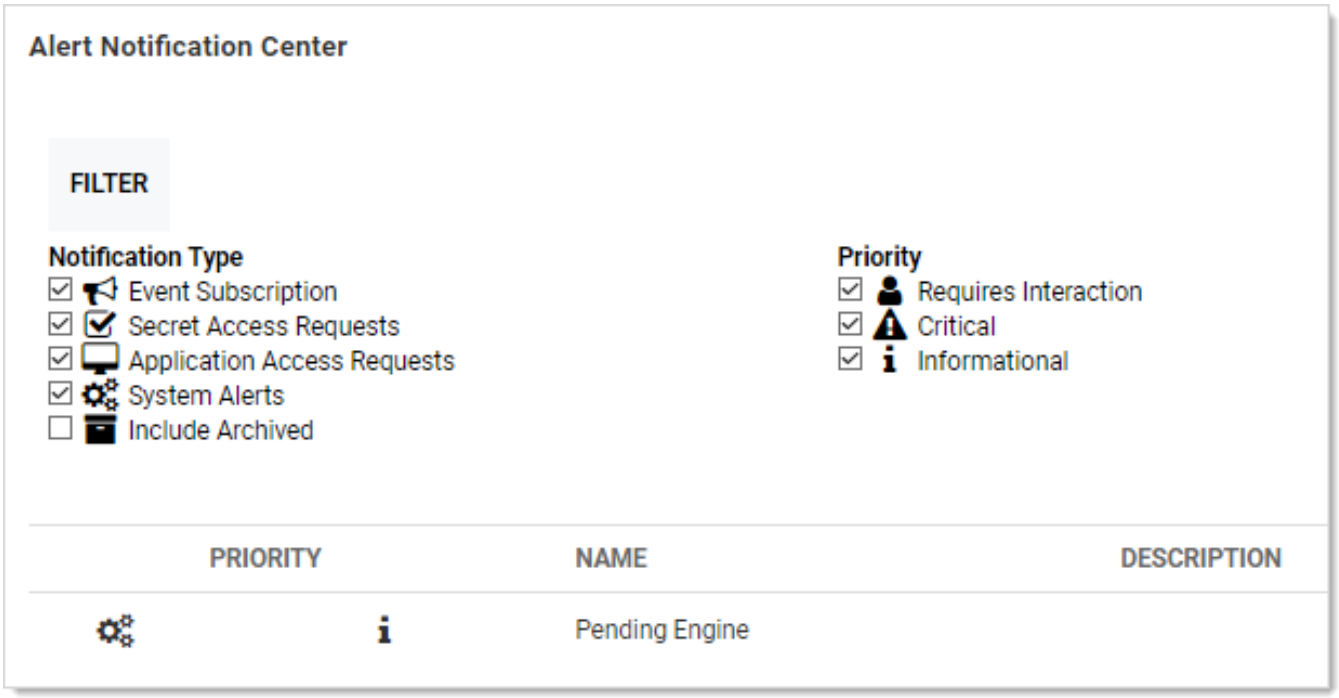
Any notifications will all be sent out at the same time, and the first response (approve or deny) will be the determinant response. A non-response will not result in either an approve or deny response.

Email Access-Request Approvals

Once a request for access to a secret has been made, approvers receive an email.

The email contains one link to the secret **Access Request Approval** page for that request in Secret Server, and five additional links to approve or deny the request if the **Allow Approval for Access from Email** configuration setting is enabled.

The approver can either click one of the links contained in the email or navigate to the **Notification Center** in the user menu within Secret Server.



If choosing the latter, in the displayed grid click the access request name. This takes you to the secret's Access Request Approval page.

From here, you can accept or deny the request as well as set an expiration date.

The requestor has access to the secret until the specified date.

Selecting the current date is the smallest window of time allowed and grants access to the end of the day.

With **Allow Approval for Access from Email** enabled, clicking one of the five additional links in the email allows access for 1, 2, 4, or 8 hours or deny the request, per the link description in the email.

> The expiration date referred to in approval requests is **not** the same as secret expiration.

Configuring Access Requests

To enable Access Request for a secret, navigate to the **Admin>Secret Policy** and select the related policy from the list.

1. Select the **Security** tab and click the **Edit** button.
2. In the **Require Approval for Access**dropdown select the related option:
   - Approval not required - no approval is required from any roles.
   - Standard exceptions - owners, editors and approvers do not need approval.
   - Owners and Approvers - owners and approvers do not need approval.

- Owners only - owners do not need approval.
- Approval always required - always require approval from all roles.

> Users need at least view access to the secret to be able to access the secret even with **Access Request** enabled. If the users do not have view permission they are unable to find the secret with search or browse.

> The email configuration settings need setting up, including valid email addresses, for the users in the approval group for emailing to work.

Requesting Access

To start the request process for access to a secret, the user must simply attempt to view the secret. The user is then sent to the Request page. In there, the user can explain the reason for the request and then click **Request Access** to submit the request.

If a member of the Approval Group either approves or denies the request (see below for details), the requestor is sent an email with the details. If approved, the requestor can access the secret via the link contained in the email.

Checkout Overview

Introduction

The Secret Server*checkout* feature forces accountability on secrets by granting exclusive access to a single user. If a secret is configured for check out, a user can then access it. If **Change Password on Check In** is turned on, after check in, Secret Server automatically forces a password change on the remote machine. No other user can access a secret while it is checked out, except unlimited administrators. This guarantees that if the remote machine is accessed using the secret, the user who had it checked out was the only one with proper credentials at that time.

> The exception to the exclusive access rule is unlimited administrators. If Unlimited Administration is enabled, users with Unlimited Administrator role permission can access checked out secrets.

> Secrets with a QuantumLock cannot be configured for check out.

Exclusive Access

Any user attempting to view a checked-out secret is directed to a notification dialog informing them when the secret is available. Secret Server automatically checks in secrets after either 30 minutes or the interval specified on the secret. Users can check in the secret earlier from the secret's page.

Checkout Expiration

Checkout expiration notifies users with a checked-out secret via their Secret Server inbox and inbox rules when a defined checkout period percentage has elapsed. For example, setting checkout expiration to 80% notifies users when 20% of the checkout interval remains.

This setting is available under "user experience" in the configuration page. To set checkout expiration:

1. Go to the **Admin** menu item and click **>>**. The Admin slideout appears.
2. Type `configuration` in the search text box and click the link when it appears below. The Configuration page appears.
3. Click the User Experience link at the bottom of the page. The User Experience page appears:
4. Click the **Edit** button.
5. Click the **Checkout Notification Threshold** spinner to select or enter a notification threshold percentage between 0 and 99. This sets what percentage of the checkout period elapses before an expiration notification occurs. That is, if set to 70, a notification is sent when 30% of the checkout period is left.
6. Click to select the **Enable Secret Check Out Extension** check box. The Max Check Out Extension Time setting appears. This sets the maximum time a secret check out can be extended in minutes. This does not include initial check out time, only the amount of time the check out is extended.
7. Click the dropdowns to set the day, hours, or minutes.
8. Click the **Save** button. The setting now appears on the read-only page.

Checking Out Secrets

Secrets with Change Password on Checkin configured now have the "Change Password Now" functionality available. This will enable the standard functionality of a password change, and the secret will also complete the automatic password change on checking in. This is to allow maintenance and testing of secrets protected in this manner, and a pending password change must be completed before the check-in process is allowed to begin in order to maintain a secure order of operations.

Each secret must be individually set to require check out:

1. From the **Secret View** page, click the **Security** tab to modify a secret's **Check Out** setting.
2. You must configure RPC before **Change Password on Check in** can be set.
3. Enable **Require Check Out** to force users to check out the secret before gaining access.
4. Enable **Change Password on Check In** to have the password change after the secret is checked in.

Checkout Hooks

Overview

In addition to changing the password on check in, secret owners can also specify administrator-created PowerShell scripts, called *hooks*, to run before or after checkout and check in. These are accessed from the **Hooks** tab of the secret, which only shows if checkout is enabled and PowerShell scripts have been created by an admin.

To specify a before- or after-checkout hook, click **Create New Hook** and specify the following settings:

- **Before/After:** Whether the PowerShell script should run before or after the event action.
- **Event Action:** The hook runs at either check in or checkout.
- **Name:** A descriptive name for the hook.
- **Description:** An extended description for the purpose of the hook.
- **PowerShell Script:** Administrator-created PowerShell script to run.

- **Arguments:** Any command line arguments to pass to the PowerShell script.
- **Stop on Failure:** If enabled, Secret Server prevents the event action if the script returns an error. For example, if "Stop on Failure" is selected for a checkout action, then Secret Server prevents the user from checking out the secret if the script fails.
- **Privileged Account:** If needed, the script can run as another secret's identity.

Checkout User Variables for Scripts

Checkout user variables for scripts are special code variables that return information about the user or automated task making the checkout request, rather than system or secret information. For example, the `$USERNAME` variable returns one or more user IDs related to a specific secret, whereas the `$SECRETSERVERUSERID` checkout user variable returns the user ID of the logged-on user or automated task.

These variables may also be useful for Active-Directory-related scripts.

The variables are:

Table: Checkout User Variables for Scripts:

Variable	User Action Returns	Automated Task Returns
`$SECRETSERVERUSERID`	Logged-on user's ID	-1
`$SECRETSERVERUSERNAME`	Logged-on user's name	"System"
`$SECRETSERVERDISPLAYNAME`	Logged-on user's display name	"System"
`$SECRETSERVEREMAILADDRESS`	Logged-on user's email address	Empty string

You can find the regular "system" variables in the"Editing Custom Commands" on page 1032 subsection of the Custom Password Changers section.

Configuring RPC on Check-in

Procedure

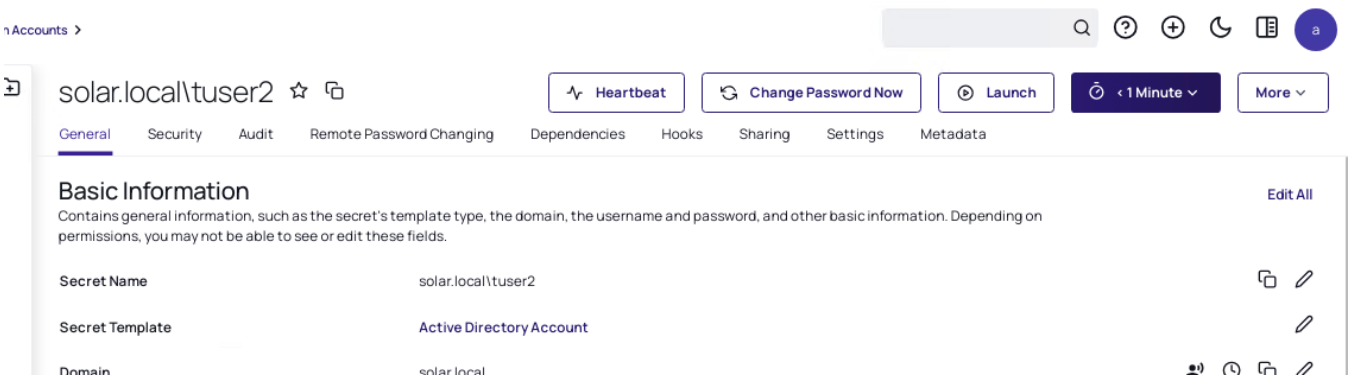
To configure password checking on check in, navigate to the **Remote Password Changing Administration** page and set **Enable Password Changing on Check In**. If RPC is turned off, enable it before configuring checkout. Once RPC and checkout are enabled, secrets can be configured for interval that specifies how long a user has exclusive secret access. The configuration of the secret is not impacted by the global level. If the secret did not have any Check-out / Check-in setup, nothing will change if you allow password change in RPC on check-in.

Manual Password Change for Checked Out Secrets

Secrets with "Change Password on Check-in" enabled have "Change Password Now" functionality. This enables the standard password-change, and the secret also completes the automatic password change on checking in. This

allows for maintenance and testing of secrets protected in this manner, and a pending password change must be completed before check-in is allowed to maintain a secure order of operations.

The functionality appears as a **Change Password Now** button on the secret:



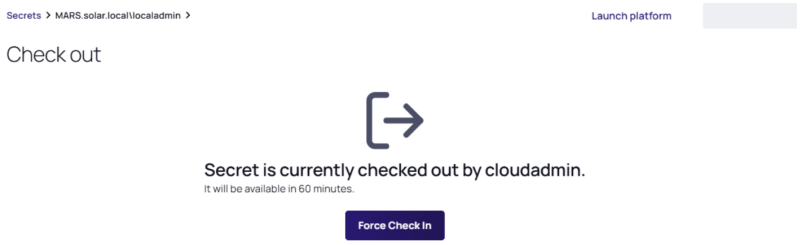
Forced Check-in

The Forced Check-in allows users with the related role permissions to forcibly check in a secret that is currently checked out by another user. This is particularly useful in scenarios where a secret is checked out and there is an urgent need for another user to access it, such as when the current user is unable to check it in.

When a user with the Force Check In role permission and ownership of the secret attempts to access it, they will see an option to force the check-in. This action automatically checks the secret out to the owner, allowing them to resolve any issues and perform a standard check-in.

To force check-in a user, checkout from the related secret - see Checking Out Secrets for details.

After that, a user should attempt to check out the same secret being logged in under his/her credentials. On this step, a user will see the Forced Check-in screen and will be able to check in the secret.



QuantumLock Overview

> QuantumLock was previously called DoubleLock.

Introduction

Secret Server's *QuantumLock* is a feature that provides an additional security layer by protecting secret data using asymmetric encryption (a public/private key pair) where the private key is a human-generated password. This

feature is independent of regular permissions, Secret Server login access, or physical access to the machine running Secret Server.

A shortcut way of thinking about QuantumLocks is as an extra password for secrets that is held by a set group of users. In addition, both the password and the group of users are reusable for other secrets. In addition, QuatumLocks future-proof our digital security infrastructure against the advancing capabilities of quantum computing.

QuantumLock is an upgrade of the earlier DoubleLock feature. Besides the name change, the difference is QuantumLock offers the option to use a quantum-safe algorithm for encapsulation to protect the private key, specifically CRYSTALS Kyber-1024, which is designed to counter the potential threat from quantum computers to current encryption methods. That threat is closer than you might think—"harvest now, decrypt later" attacks steal encrypted data now for later decryption by quantum computers.

> The private key being protected by Kyber-1024 is a human-generated user password. Once encapsulated (encrypted) that password is called a *ciphertext*. Once that private key is decapsulated (decrypted), it is used by a symmetric algorithm, such as AES-256, to decrypt the protected data. This two-step process (key encapsulation followed by symmetric encryption) applies the strengths of both asymmetric (for secure, fast key exchange) and symmetric cryptography (for efficient message encryption), providing a robust, efficient method for secure communication.

Today's asymmetric-key encapsulation methods, such as RSA (Rivest-Shamir-Adleman) and ECC (Elliptic Curve Cryptography), rely on the difficulty of solving mathematical problems, such as factoring large numbers or solving discrete logarithms, with classical computers. However, quantum computers, which operate on principles of quantum mechanics, could solve these problems much faster, rendering these encapsulation methods vulnerable.

Quantum-safe or post-quantum (PQ) algorithms are cryptographic methods that are believed to be secure against quantum computer attacks. They are based on mathematical problems that are considered difficult for both classical and quantum computers to solve, ensuring the security of encapsulated keys, even in the era of quantum computing. PQ algorithm types include lattice-based cryptography, hash-based cryptography, code-based cryptography, and multivariate polynomial cryptography, among others.

> Kyber-1024's resistance to quantum attacks is, by design, approximately the same as AES-256's resistance to conventional computer attacks.

The PQ cryptographic algorithm Kyber-1024 is specifically designed for key-encapsulation mechanisms (KEM), a process where a key is encapsulated (encrypted) with a public key, sent over an insecure channel, and then decrypted with a private key. Typically a symmetric key algorithm, such as AES-256 (Advanced Encryption Standard) , is used for encrypting the message content itself because symmetric-key algorithms are faster and more efficient for large amounts of data than asymmetric-key algorithms. Algorithms like AES are examples of symmetric cryptography.

> You might wonder how well protected the encrypted data at rest is to quantum-computer attacks. Symmetric algorithms, such as AES, are PQ if their keys are sufficiently long, which is not an issue with their typical use cases.

The Kyber algorithm comes in different security levels, and Kyber-1024 specifically aims at security roughly equivalent to AES-256, with a private key size of 3168, a public key size of 1568, and a ciphertext size of 1568. It is part of Cryptographic Suite for Algebraic Lattices (CRYSTALS), which is a package submitted to the NIST post-quantum standardization effort.

Comparing RSA-2048 to Kyber-1024

QuantumLock without the QuantumLock feature enabled is essentially the same as its predecessor, DoubleLock, and relies on RSA-2048 for key encapsulation.

Comparing the encryption, decryption, and key-generation speeds of RSA-2048 and Kyber-1024 involves understanding the efficiency of these algorithms under practical implementations. The specific speeds can vary based on the software and hardware used for the implementation, but here is a general overview based on their cryptographic principles and typical use cases.

RSA-2048

- Key Generation: RSA-2048 key generation is relatively slow because it involves finding two large prime numbers and calculating their product along with other related mathematical operations. This process is computationally intensive.
- Encapsulation: RSA-2048 encryption is faster than its key generation. However, compared to Kyber-1024, RSA-2048 encryption is usually slower because it involves modular exponentiation, which is a heavy operation especially for large key sizes like 2048 bits.
- Decapsulation: RSA-2048 decryption is also computationally intensive, similar to encryption, because it requires modular exponentiation. RSA decryption is generally slower than encryption due to the nature of the private key operations.

Kyber-1024

- Key Generation: Kyber-1024 generally has faster key generation than RSA-2048. This efficiency comes from its use of lattice-based cryptography, which involves operations on vectors and matrices that are more efficient than the prime number operations in RSA.
- Encapsulation: Kyber-1024 is designed for fast encryption operations. It uses simple arithmetic operations on small integers, making it very efficient and faster than RSA-2048 encryption.
- Decapsulation: Like its encryption, Kyber-1024 decryption is also fast and efficient. The algorithm benefits from the same lattice-based operations, optimized for quick decryption times.

Summary

- RSA-2048 is generally slower across all three operations compared to Kyber-1024. The difference in speed is primarily due to RSA's reliance on large prime numbers and modular arithmetic, which are computationally heavier, especially as key sizes increase to improve security.
- Kyber-1024, being a post-quantum algorithm designed with efficiency in mind, uses lattice-based cryptography that allows for quicker key generation, encryption, and decryption operations. This makes it particularly suitable for environments where speed and efficiency are critical.

While Kyber-1024 offers advantages in speed and quantum resistance, the choice between RSA-2048 and Kyber-1024 (or any cryptographic algorithm) depends on the specific security requirements, computational resources, and threat models relevant to the application in question.

When to Use QuantumLock

Both Kyber-1024 and RSA-2048 QuantumLocks

Enabling QuantumLock (or the earlier doublelock) on any secret only grants users access if they have access to the QuantumLock and enter their QuantumLock password. Enabling QuantumLock disables the RPC features for the secret. It also prevents heartbeat. Thus, **QuantumLock should not be used for secrets that require a password rotation or heartbeat check**.

QuantumLock use cases include:

- Global admin passwords
- Root account passwords
- Bank account passwords
- PINs, Social Security numbers, or other personal information.

When users protect secrets with QuantumLock, only that user has access to the secret. If multiple users are in a group, members have access to the secret, but each will have their own unique QuantumLock.

Even an administrator or a user with unlimited admin privileges cannot recover a QuantumLocked secret if the user forgets his or her password. If there is a single user in a QuantumLock group and that user account is deleted, those secrets will not be accessible by anyone! Thus, we recommend adding at least two users to the group to be safe.

Kyber-1024 QuantumLocks

Enabling the PQ feature of QuantumLock depends on your circumstances. NIST believes quantum computers could break current public key encryption as early as 2025 or as late as 2030. It might be wise to apply it sooner rather than later for long-term data and devices. However, Kyber-1024 is not yet an industry standard, and RSA-2048 is currently uncrackable.

Note that users cannot remove their own QuantumLocks. When attempting to remove their QuantumLocks, users are presented with a 500 error, which means that this is not a supported operation.

QuantumLock Objects and Relationships

QuantumLock was previously called doublelock.

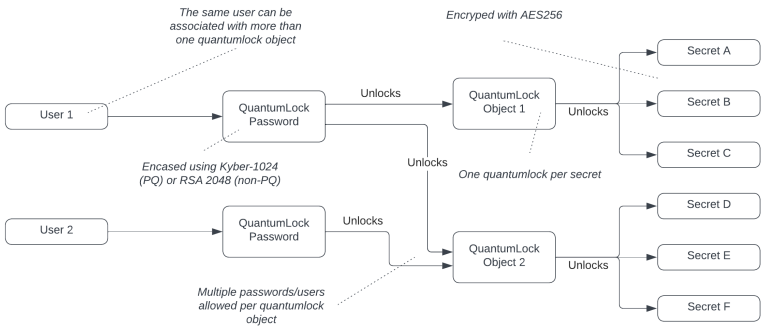
The QuantumLock system is a group of interrelated objects (see the following diagram):

- **QuantumLock user group:** A set of Secret Server users that have access to a single QuantumLock. This is not an object per se—it is the set of users assigned to a QuantumLock object. Each user has a separate password via the QuantumLock password object.
- **QuantumLock object:** A named object that is associated with one or more secrets and one or more users. QuantumLock objects, or simply *QuantumLocks*, secure one or more secrets, and one or more QuantumLock password objects provide access to it. The secrets themselves are encrypted with AES256, and are unlocked using the symmetric key provided by the QuantumLock object.
- **QuantumLock password object:** An encrypted asymmetric password that is associated with one user. The same QuantumLock password object, or simply *QuantumLock password*, is used for all QuantumLocks to which a user has access. Other users associated with the same QuantumLock have their own QuantumLock passwords. Once a user is assigned to a QuantumLock, that user has access to any secret using that QuantumLock, using the user's single QuantumLock password.

  A QuantumLock password has nothing to do with the user's Secret Server access password.

- **Secret:** A secret that has a single QuantumLock assigned to it. Multiple secrets can have the same one assigned to them. The secret is encrypted with AES256, with its symmetric key provided by the QuantumLock object, which is itself secured by one or more QuantumLock passwords, which are encased by Kyber-1024 or RSA-2048.
- **User:** A Secret Server user, which can have a single QuantumLock password assigned to it. Users are assigned to QuantumLock objects.

Figure: QuantumLock Object Relationships



Administering QuantumLocks

Administering QuantumLocks in Secret Server involves managing an advanced security feature designed to protect secret data using asymmetric encryption, where the private key is a human-generated password. QuantumLocks, previously known as DoubleLocks, provide an additional layer of security independent of regular permissions and Secret Server login access. This feature is particularly useful for securing highly sensitive data against the potential future threats posed by quantum computing. QuantumLocks utilize a quantum-safe algorithm, such as CRYSTALS Kyber-1024, to encapsulate the private key, ensuring robust protection against "harvest now, decrypt later" attacks. Administrators must carefully manage QuantumLock user groups, ensuring that multiple users are assigned to prevent data loss if a single user forgets their password or is deleted. This feature is best suited for protecting global

admin passwords, root account passwords, and other critical information that does not require frequent password rotation or heartbeat checks.

Assigning Users to QuantumLocks (Administrators)

> Ensure you have the Administer DoubleLock Keys permission prior to performing this procedure. DoubleLock was the earlier name for QuantumLock.

1. Click **Administration** in the main menu. The Administration page appears.
2. Click the **QuantumLock** link in the **Diagnostics, Logs, Security** section. The QuantumLock Management page appears.
3. Click the desired QuantumLock. Its page appears.
4. Click the **Add or Remove** link in the **Assign Users** section. A table of users with a QuantumLock password appears.
5. If you do not immediately see the desired user, type the user's name in the search text box. The matching users appear below the search text box.
6. Click to select the dropdown list to the right of the search box to limit the search to assigned or unassigned users. These are the users that are already assigned or not to the QuantumLock object (user group).
7. Click to select the check box next to the desired users.
8. Click the **Save** button.

Configuring QuantumLocks (Administrators)

> Ensure you have the Administer DoubleLock Keys permission prior to performing this procedure.

As an admin, to use QuantumLocks on a secret, you must first create complete these steps for a new QuantumLock:

1. One time: Enable the "quantum safe encryption" feature of QuantumLock. See "Enabling Quantum Safe Encryption(Administrators)" on page 1087.
2. One time: Create a QuantumLock password (one time per user). This is automatically required of you when you create a QuantumLock or access a secret with an existing one (that somebody else assigned to you). You can also create one manually ahead of time. See "Creating QuantumLocks (Administrators)" on the next page.
3. One time: Create a QuantumLock, which can be used on multiple secrets by multiple users. See "Creating QuantumLocks (Administrators)" on the next page.
4. One or more times: Assign the QuantumLock to a secret or secret template. See "Enabling QuantumLocks on Secrets" on page 1088
5. One time per user: Assign the user to that QuantumLock. Users without an existing QuantumLock password are required to create one. See "Creating a QuantumLock Password" on page 1087 or "Assigning Users to QuantumLocks (Administrators)" above

6. Unlimited times: A user unlocks the QuantumLock with his or her QuantumLock password, which in turn gives the user access to the secret associated with the QuantumLock.

> When a user initially unlocks the QuantumLock, they are not required to reenter their password for one hour after they initially unlocked it. After an hour, the QuantumLock returns to requiring the password. The next time they unlock it, the one-hour "waiver window" starts anew.

Creating QuantumLocks (Administrators)

> Ensure you have the Administer DoubleLock Keys permission prior to performing this procedure.

> We recommend reading "QuantumLock Overview" on page 1080 and "QuantumLock Objects and Relationships" on page 1083 before proceeding.

1. Click **Administration** in the main menu. The Administration page appears.
2. Click the **QuantumLock** link in the **Diagnostics, Logs, Security** section. The QuantumLock Management page appears.
3. Click the **Create New QuantumLock** button. If you have never created a QuantumLock before, the Create QuantumLock Password popup appears, and you have to create a QuantumLock password first. Otherwise, you go directly to the Create New QuantumLock page because you already have a QuantumLock password in the system.
4. Type your QuantumLock password in the **Password** and **Confirm** text boxes.

   > It is critical that you remember or securely store this password. It cannot be recovered.

   > Because it is a secondary password, your QuantumLock password does not have to (but can) meet the same strong requirements as regular Secret Server passwords (as defined by your admin). The only requirement is it must be between 8 and 500 characters long, inclusive.

   > A new QuantumLock and QuantumLock password are created together. In fact, it is impossible to create a QuantumLock password without immediately assigning it to a QuantumLock or vice versa. For an existing QuantumLock, you are assigned access to it by its creator. Upon first accessing it, you must create *your* QuantumLock password for it. At least one other user will already have created their password for the same QuantumLock–the creator plus anybody else they granted access to.

5. Click to select the desired key encapsulation algorithm. See "QuantumLock Overview" on page 1080 for details.
6. Click the **Create** button. The Create New QuantumLock popup appears.
7. Type the new QuantumLock's name in the **Name** text box.
8. Click the **Create New QuantumLock** button. The new QuantumLock's page appears. See the following for more information:

- "Assigning Users to QuantumLocks (Administrators)" on page 1085
- "Configuring QuantumLocks (Administrators)" on page 1085

Enabling Quantum Safe Encryption(Administrators)

By default, the Quantum Safe Encryption feature of QuantumLock is disabled. That is RSA-2048 and not Kyber-1024 is used by default. To enable the feature:

> Ensure you have the Administer DoubleLock Keys permission prior to performing this procedure.

> We recommend reading "QuantumLock Overview" on page 1080 and "QuantumLock Objects and Relationships" on page 1083 before proceeding.

> These instructions are only applicable if the Quantum Safe Encryption was not previously enabled.

1. Search for Quantum Safe Encryption in the Configuration search.
2. Check the **Enable Quantum Safe** box.
3. Go to **Quantum Lock Management**.
4. Update your password to use Kyber.

Using QuantumLocks

Using QuantumLocks in Secret Server involves leveraging an additional security layer to protect secret data through asymmetric encryption, where the private key is a human-generated password. This feature, previously known as DoubleLock, is independent of regular permissions and Secret Server login access. QuantumLocks are particularly useful for securing highly sensitive data against future quantum computing threats. When a secret is protected with a QuantumLock, only users with access to the QuantumLock and the corresponding password can decrypt the secret. This ensures that even if Secret Server is compromised, the secrets remain secure. QuantumLocks can be applied to various types of sensitive information, such as global admin passwords, root account passwords, and personal information. However, it is important to note that enabling QuantumLock disables certain features like Remote Password Changing (RPC) and heartbeat, making it unsuitable for secrets requiring frequent password rotations.

Creating a QuantumLock Password

1. Click the user icon at the top right of Secret Server and select **User Preferences**. Your User Preferences page appears.
2. Click the **Change QuantumLock Password** link. The Enter QuantumLock Password popup appears.

   > You cannot create a QuantumLock password until you are associated with a QuantumLock. When you access your first QuantumLock, you are prompted to create a password.

3. Type your desired doublelock password in the **Password** and **Confirm Password** text boxes.

> It is critical that you remember or securely store this password. It cannot be recovered.

4. Click the **Change Password** button. The password is created.

Enabling QuantumLocks on Secrets

1. Navigate to the secret you wish to QuantumLock by clicking **Secrets** on the main menu.
2. Either drill down to the desired secret in the folders on the main menu, or click the secret in the **All Secrets** table to arrive at the secret's page:
3. Click the **Security** tab.
4. Click the pencil icon to the right of **Enable QuantumLock** in the **Other Security** section. The Create QuantumLock Password popup appears.
5. Click to select the **Enable QuantumLock** check box. The QuantumLock dropdown list appears.
6. Click to select the QuantumLock you created earlier.

> Enabling QuantumLock on this secret only grants users access if they have access to the QuantumLock and enter their QuantumLock password. Enabling QuantumLock disables the RPC features for the secret.

7. Click the **Save** button. The QuantumLock is now enforced for the secret.

Recovering QuantumLock Passwords

Introduction

Because users with access to a given QuantumLock each have their own separate password. If you forget your QuantumLock password, you cannot simply ask another person using that QuantumLock for the password. Instead, one of the other users must reassign you to the QuantumLock, and you must create a new password. Because your QuantumLock password is the same for all your QuantumLocks, you must repeat the process for all those QuantumLocks. If no other QuantumLock users are available to assign you to the QuantumLock (there only one associated QuantumLock password–the one you cannot remember), you are out of luck, and the secret will be destroyed when you receive a new QuantumLock password.

> Because of this, we strongly recommend always having more than one user per QuantumLock if at all possible.

Recovery Procedure

When you forget your QuantumLock passwords, there are multiple steps and considerations, and loss may or may not result:

1. When you forget your QuantumLock password, you typically come to that realization when attempting to access a secret protected by that QuantumLock.
2. Click the **Forgot QuantumLock Password?** link. The Reset QuantumLock Password page appears.

3. At this stage, there are two possibilities:
   - You are the only one with access to the QuantumLocked secret: When you reset the QuantumLock password, the secret and its data is deleted. **This is permanent**.
   - Others have access to the secret via that QuantumLock: You can reset the QuantumLock, and you lose access to the secret, but it is not deleted. You must ask one of those other users to re-assign you to the QuantumLock after you reset it.
4. Type your main Secret Server password in the **Login Password** text box.
5. Click the **Reset QuantumLock Password** button. The password is reset, and if you are the only one with access to the QuantumLock, the secret is deleted.
6. (Optional) Ask one of the others with the QuantumLock password to re-assign you to the QuantumLock.

Setting up QuantumLocks (for users)

To use QuantumLocks, you must:

- Set up a QuantumLock password. See "Creating a QuantumLock Password" on page 1087.
- Have an admin:
  - Create a QuantumLock object.
  - Assign you to that QuantumLock's user group.
  - If you are the only person who should have access to the user group, remove themselves from the group.
  - Tell you the name of the user group (the QuantumLock object name).
- Enable QuantumLock on one or more secrets, assigning a QuantumLock object (group name) to each. See "Enabling QuantumLocks on Secrets" on the previous page.

Workflow Overview

Starting in 10.6, Secret Server introduced *access-request workflows*. These allow users to build more complex interactions based on events within Secret Server than currently possible. The first release of workflows offers access requests. Workflow templates define the series of steps and reviewers required for an access request. You can assign workflows to secrets or secret policies.

With Access-Request Workflow Templates, you can:

- Require that multiple people approve a request before access is granted
- Require multiple workflow steps, each with different reviewers and number of required approvers, if desired.
- Select "Owners" as a review group

Access Requests already existed in Secret Server, but with 10.6 they become much more powerful. Previously, if access requests enabled on a secret, requests were granted after a single reviewer approved the request. Now, approval workflows can require multiple approvers, and multiple approval levels.

Note that the Approval Workflow is available only for Professional (with Licensing Add-On) and Platinum Secret Server users. See the Secret Server Feature Chart for more details.

Multi-Level Workflow

The original access requests are one level or step–anyone approving approves the request–no other input is required. Workflows allow up to 15 approval steps where approval by reviews in step 1 moves the request to step 2, approval at step 2 moves it to step 3 and so forth. Denial at any step denies the request.

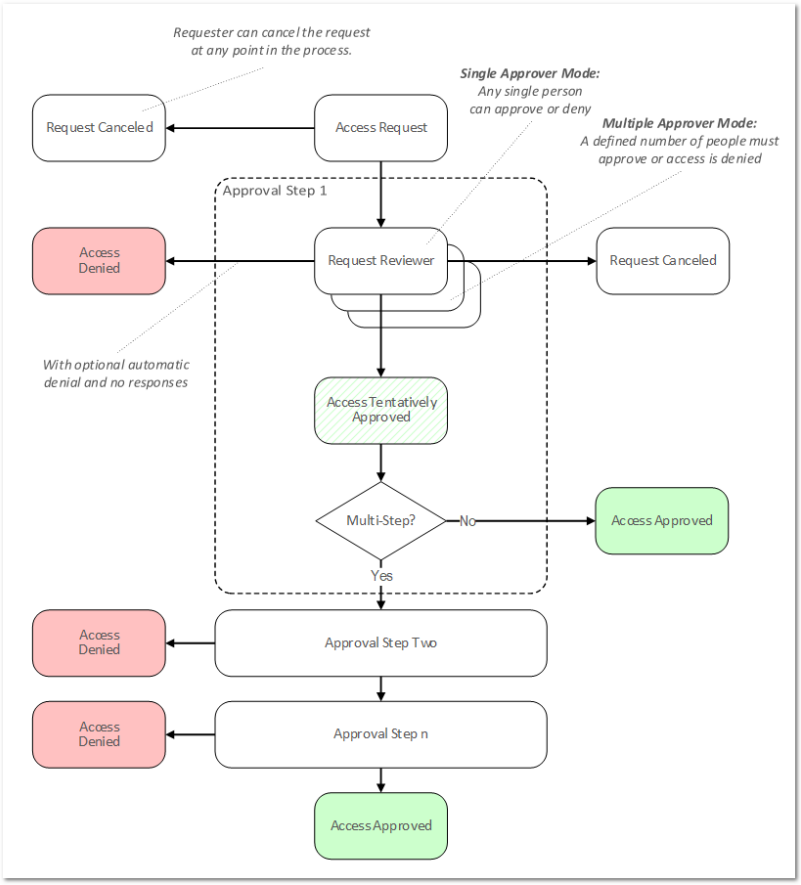
Multiple Approvers to Advance

The new workflow feature can be configured where one approver at a given step is not enough. In effect, approvers in each step can "vote" for approval–you stipulate how many approvers at a step must approve for the approval to move on to the next step.

Approval Process Workflow

The following diagram is the entire process summarized:

Figure: The Approval Process Workflow



Workflow Versus Basic Access Requests

In general, "simple access requests," the only type available to older versions of Secret Server, are the same as a one-step stepped approval. The major exception is that with stepped requests, once a workflow access request has been approved, denied, or canceled, its status cannot be changed. In contrast, simple, non-workflow, access requests retain the original behavior of allowing a request to be approved after it has been denied or denied after it has been approved.

Workflow Step Timeout

Release notes

You can configure workflow steps to time out after a specified number of minutes. Workflow administrators can define approval workflows that notify a different set of users if a step in the workflow is not responded to within a specified time period.

Applications include:

- Improving responsiveness to access requests on time-sensitive secrets by moving the request to another step if not responded to quickly.
- Overflowing to a different region so a secret can be accessed by users in different time zones with different sets of support personnel responding.

Timed out access requests automatically advance to the next step in the workflow. The last step of the workflow cannot time out and must be approved to access the secret.

You can apply multiple timeouts to create workflows that can cascade through multiple steps if the previous steps do not receive the required approvals.

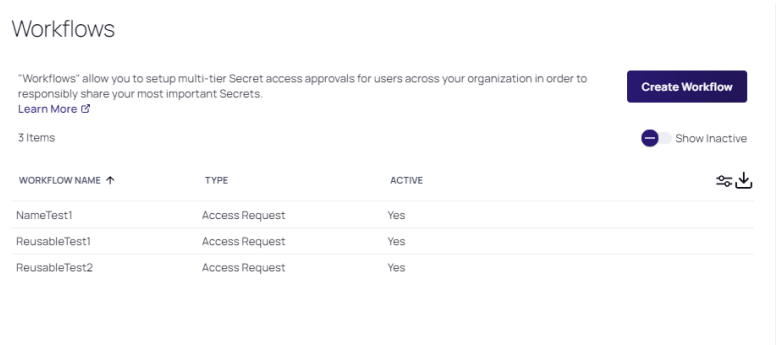
Denying or canceling a request in any step stops the workflow and stops any time out to the next step.

Important: Once a workflow step times out, only the reviewers in the next step can approve the request. If you want reviewers from the initial step to respond after the initial request has timed out, add them to the reviewers of the next step.

Accessing the Workflow Designer

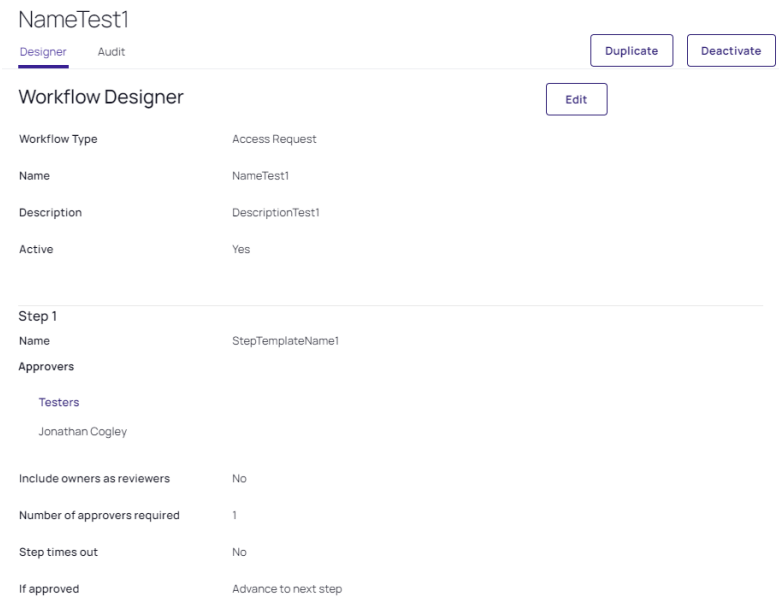
To access workflows:

1. Go to **Admin > Workflows**. The Workflows page appears:



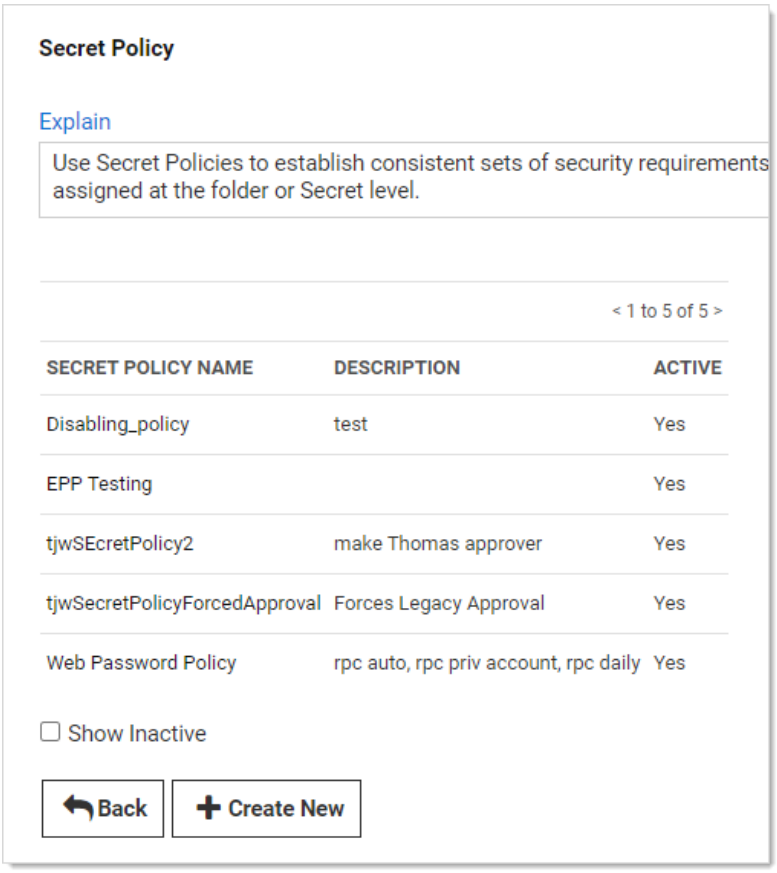
The page lists all active workflows.

2. (Optional) Click to enable the **Show Inactive** toggle button, under the **Create Workflow** button, to show both active and inactive templates. When the toggle button is disabled, it only shows active workflows.
3. Click any workflow in the list to go to the designer page for that workflow:

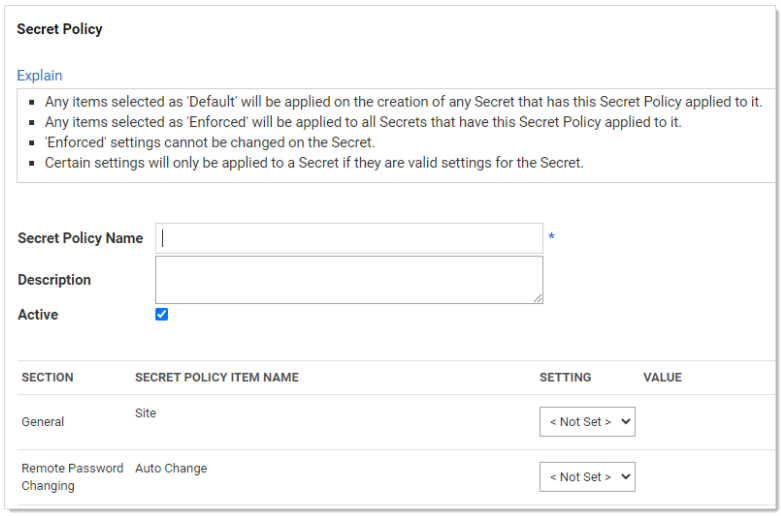


Assigning Workflows to Secret Policies

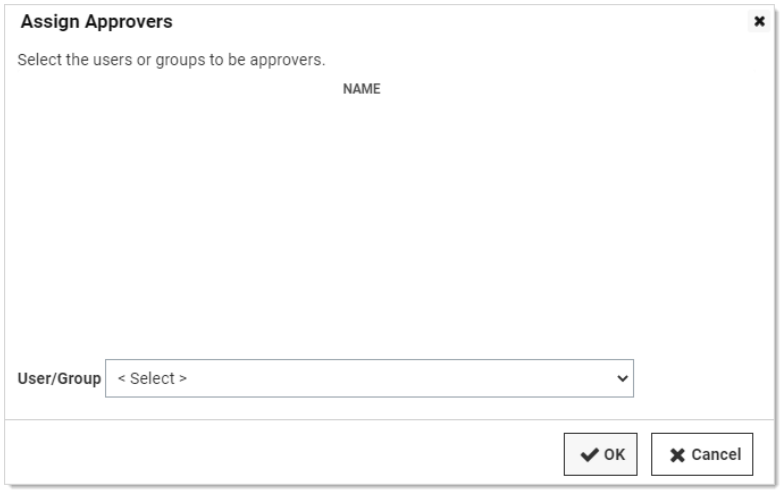
1. Click **Admin > Secret Policy**. The Secret Policy page appears:



2. For this instruction, we are going to create a new policy.
3. Click the **+ Create New** button. Another Secret Policy page appears:

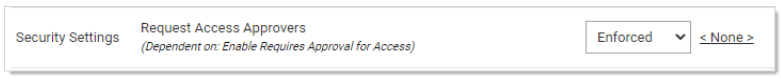


4. Type the new policy name in the **Secret Policy Name** text box.
5. Scroll down the page to the **Security Settings** section of the unlabeled table.
6. Click the **Enable Requires Approval for Access** list and select **Enforced**.
7. Click to select the check box next to the list. The Assign Approvers popup page appears:



8. Click the **Cancel** button. The Request Access Approvers setting become enabled:

> You cannot set approvers and use a workflow at the same time. The intent of the next few instructions is avoid attempting to do so, which causes an error.



9. Click the **Request Access Approvers** list and select **Not Set**.

10. Click the **Request Access Workflow** list and select **Enforced**. A new list appears alongside:

11. Click the new unlabeled list and select the access template workflow to associate with the policy.
12. Click the **Save** button at the bottom of the page. The policy is now available for assignment to secrets and folders, just like any other policy.

Creating New Workflows

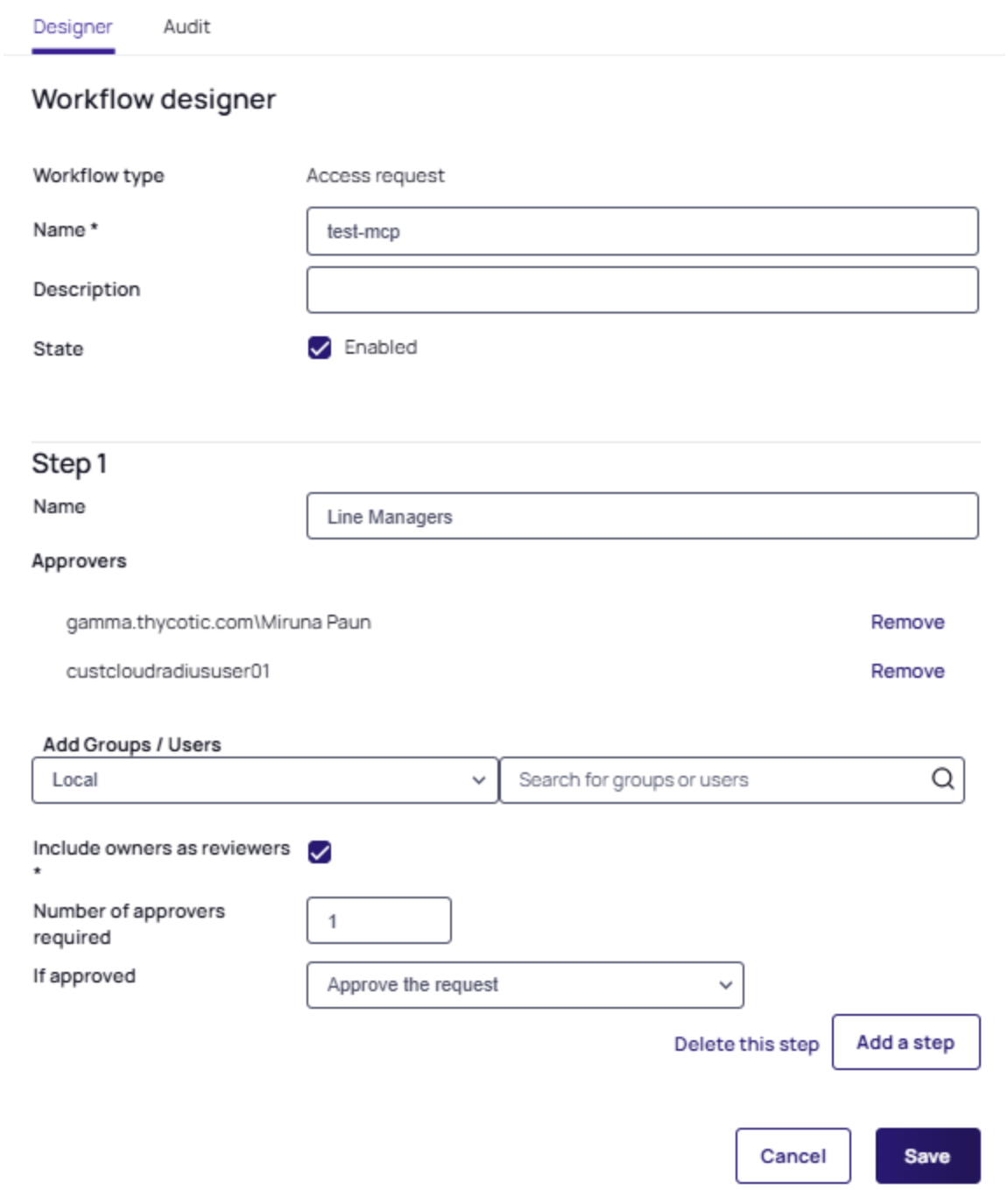
Task 1: Access the Workflow Designer

1. Search for **Workflow**. The Workflows page appears.
2. Select the **Create Workflow** button. The Create Workflow popup appears.
3. Type in the **Workflow name** and **Description** (optional) text boxes, then choose a **Workflow type**.
4. Click the **Create Workflow** button. The Workflow Designer tab loads with the warning that the workflow needs to be activated (enabled) for it to work.

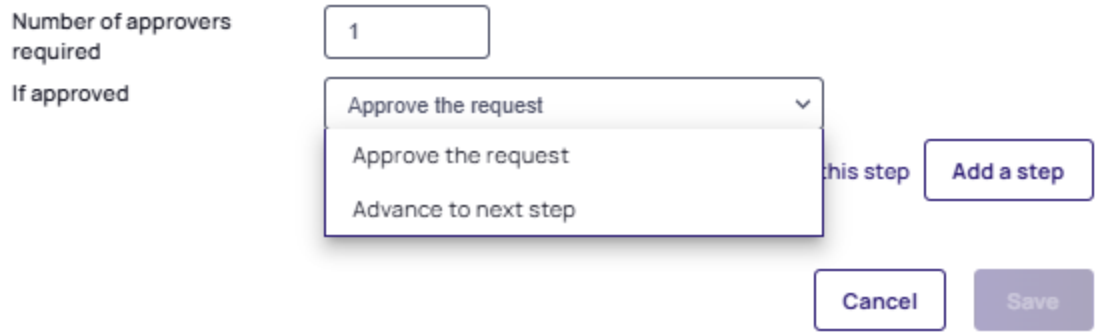
> A new workflow has only one empty step by default.

Task 2: Set up the First Step

1. On the workflow **Designer** tab, click to select the **State** checkbox as **Enabled**.
2. (Optional) Change the default text for Step 1 in the **Name** text box, for example "Line Managers".
3. In the **Add Groups / Users** dropdown list, select the domain where you want to get your approvers from.
4. Type the name of the user or group you desire as an approver in the unlabeled search box to the right. Options appear in the dropdown.
5. Click the desired user or group. The user is added to the **Approvers** section automatically, under the Step 1 name field.
6. Repeat as desired.
7. (Optional) To automatically include the owner of the secret the template is assigned to, select the **Include owners as reviewers** checkbox:



8. If you wish to have multiple approvers required on the step, edit the **Number of approvers required** text box. Otherwise, leave it set to the default 1.
9. Click the **If approved** dropdown list to select what to do next:



10. Once selected, click **Save**.

Task 3: (Optional) Add More Steps

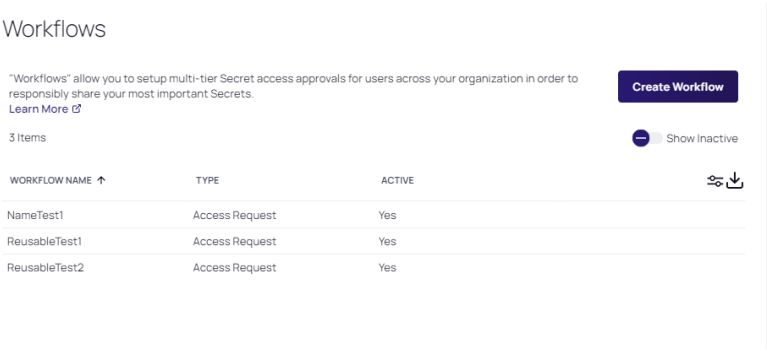
> There are situations where you might want to have only one workflow step. Most Workflows, however, provide options to require multiple approvers or have owners as approvers, which are not available for simple access requests.

1. Select the **Add a Step** button. A new step appears.
2. Repeat the process mentioned in Task 2, keep adding steps as needed.
3. Click the **Save** button to create or save the workflow. The template exits editable mode.

Deleting Workflows

To delete a workflow:

1. Go to **Admin > Workflows**. The Workflows page appears:



   The page lists all active workflows.
2. (Optional) Click to enable the **Show Inactive** toggle button, under the **Create Workflow** button, to show both active and inactive templates. When the toggle button is disabled, it only shows active workflows.
3. Click the workflow to delete in the list to go to the designer page for that workflow (not shown).

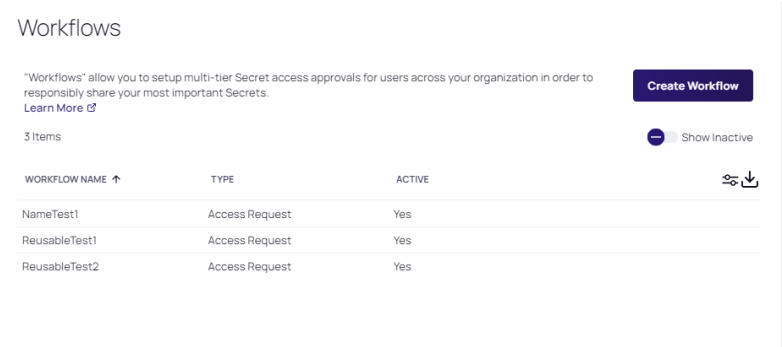
4. Click the **Delete** button. A confirmation popup page appears.
5. Click the **Yes, Delete** button.

> Because workflows based on the template may still be in play, the template is not completely deleted. Instead, it is inactivated. You can reactivate the template later. See "Accessing the Workflow Designer" on page 1091.

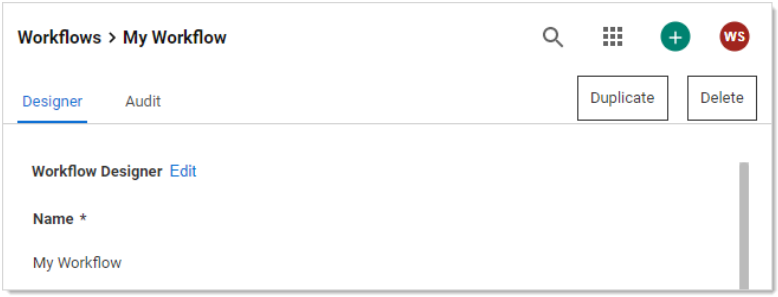
Duplicating Workflows

If you need to create a new workflow that is like one your already have, you can save time by copying the similar template and then making the any changes:

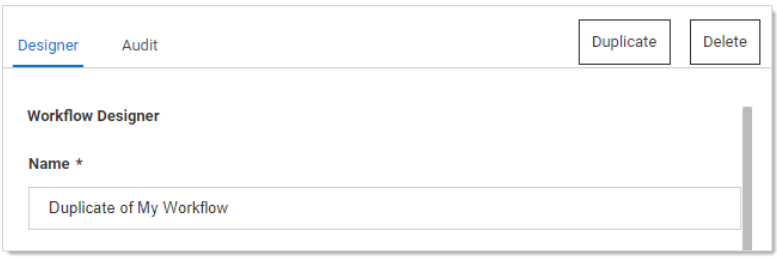
1. Go to **Admin > Workflows**. The Workflows page appears:



   The page lists all active workflows.
2. (Optional) Click to enable the **Show Inactive** toggle button, under the **Create Workflow** button, to show both active and inactive templates. When the toggle button is disabled, it only shows active workflows.
3. Click the workflow you want to copy in the **Workflow Templates** table. That template appears:



4. Click the **Duplicate** button. The new template appears, filled in the same as the original but with a "Duplicate of" name:

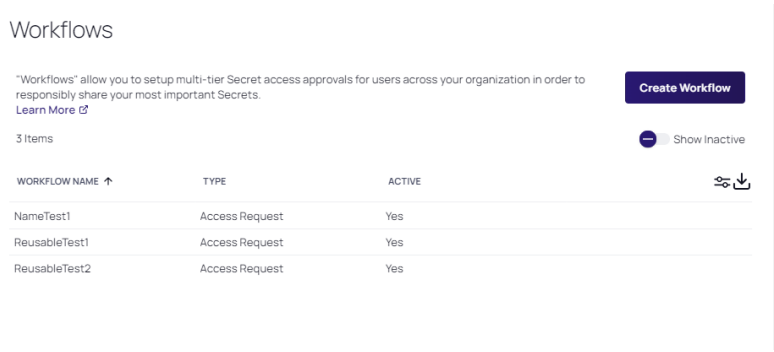


5. Change the name and edit as desired.
6. Click the **Save** button when finished.

Editing Workflows

To edit the template:

1. Go to **Admin > Workflows**. The Workflows page appears:



   The page lists all active workflows.
2. (Optional) Click to enable the **Show Inactive** toggle button, under the **Create Workflow** button, to show both active and inactive templates. When the toggle button is disabled, it only shows active workflows.
3. Click the workflow to edit in the list to go to the designer page for that workflow (not shown).
4. At this stage the process is nearly identical to creating a new workflow. The only difference is many of the parameters and additional steps are already completed. Change them as desired. If you want to eliminate an entire step, click the **Delete This Step** link for that step.

> You cannot make any changes to the behavior of a workflow if there are active requests using that template without canceling those requests. An active request is any unexpired request that has not been approved, denied, or canceled by the user. If you do make an alteration, any requests are canceled and those affected are notified by email so they can resubmit their requests. Any user editing the template is notified when he or she tries to save changes on the canceled request.

Workflow Design Best Practices

Consider the following when setting up an access-request workflow:

- Use multiple-step approval workflows when you need to have different people (such as different departments) sign off on an approval request.
- We do not recommend assigning equally important approvers or groups to multiple steps. Having a single step with multiple approvers works better. Remember, steps are best used for hierarchical approval--an approval chain.
- A reviewer can only respond to a request once. If you have the same user as a reviewer in multiple steps, that approver cannot respond if he or she already responded on an earlier step. In addition, the reviewer's earlier approval does **not** count towards the number of approvals required in later steps. Thus, if you want to assign the same user as a reviewer in multiple steps, make sure that you have enough reviewers in each step to approve without that user.
- A well-crafted workflow design ensures there are enough approvers in a group to satisfy the multiple approver (x of n reviewers must approve) requirement, but group membership can change after the workflow is created. Thus, if you remove members from groups used by workflows, ensure there are still enough members in those groups to approve requests.
- Once a workflow step times out, only the reviewers in the next step can approve the request. If you want reviewers from the initial step to respond after the initial request has timed out, add them to the reviewers of the next step.

Secret Import and Export Overview

Introduction

Secrets are imported or exported as a comma-separated-value (CSV) file or as XML:

- The CSV file is easily read and edited in Excel or other spreadsheet application. The file is grouped by secret template and each cluster of secrets has a header row that contains the template text-entry field names followed by all exported secrets based on that template.
- The XML file is useful for migrating data from one Secret Server installation to another or even from a third-party application to Secret Server.

Secrets are exported in the exact same structure as a secret Import.

This topic has three subtopics:

- "Exporting Secrets" on page 1109
- "Importing Secrets" on page 1111
- "Secret Server Migration Tool" on page 1117

What Gets Imported or Exported

Import and export include:

- Folders (and their permissions)
- Secret templates
- Secrets (and their permissions)

The import or export does **not** include users, groups, launchers, configuration, and others.

Folders and secret templates are only exportable from Secret Server 10.0 and later.

To ensure permissions are applied correctly, you must recreate your users and groups on the target Secret Server before importing.

The following secret template settings **are** transferred with the export or import:

- Edit requires
- Field slug names
- Hide on view
- Is required?
- Keep secret name history
- One-time password settings
- Secret template icons
- Type descriptions
- Validate password requirements on create or edit

The following secret template settings are **not** transferred:

- Associated secrets
- Launcher settings
- Password changing settings
- Session recording enabled

Migrating to and from Secret Server Cloud

If you use XML import and export to migrate from Secret Server on-premises to cloud, the major release version (x.x) must be the same. Otherwise, you need to upgrade before you can migrate. Additionally, the **Allow Duplicate Secret Names** check box on the **General** tab of the **Admin Configuration** page should be disabled in Secret Server Cloud before importing.

It is not recommended to use XML import and export to transfer between on-premises and cloud editions. Instead, you should use professional services, or leverage the API instead.

Automatic Secret Export

This feature allows you to automatically export secrets on a schedule to an external location in an encrypted, password-protected archive.

The secret export settings and XML export format matches the existing Export/Import tool (Admin > Export / Import). This feature lets you automate that export process.

To access this feature, your user must have at least one automatic export permission where you can then find it at **Admin > Automatic Export**.

The export is performed using the permissions of the user last that set up the automatic export, this means only secrets that user can access can be exported.

All actions, successful or not, related to this feature are audited and logged.

Export Process

> The actual export of secrets to XML is exactly the same as the standard Export / Import tool—only the triggering differs.

The automatic export follows this process in order:

1. Either a user clicks the Run Export button on the Automatic Export tab or the configured frequency days have passed.

   > Users can manually run the automatic export to determine if the export performs as desired.

2. Secret Server determines if the user who set up the automatic export has either the Run Automatic Export or Administer Run Automatic permission.
3. Secret Server exports secrets to XML, subject to your export settings.
4. Secret Server compresses the XML export into an encrypted, password-protected ZIP archive. The filename for both the XML file and the zipped XML file includes the export date and time—both are named the same, except the file extension. The encryption is 256-bit AES, and the password comes from the Export Password configuration setting.
5. Secret Server saves the archive to the export path (on-premises customers) or to cloud storage (cloud customers).
6. Secret Server logs the results, successful or not, noting any errors. You can view the logs on the Automatic Export Log tab.

Considerations and Settings

Configurations and Returned Data

The Automatic Export page shows the following information and configurations:

Those unique to automatic exports:

- **Enabled**: Whether the feature is enabled.
- **Last Exported:** The last time an automatic export successfully ran.
- **Export Path:** The path the export archives are saved to.

  > On-premises Secret Server customers must have write permissions to the directory.

- **Export User:** The user the secret export runs as. Thus, only secrets this user has access to can be exported. This setting is updated automatically to the last user who saved automatic export configuration changes.
- **Frequency (Days):** Number of days between automatic exports.

Those in common with the secret export tool:

- **Export Child Folders:** Whether the export should include child folders, performing a recursive export.
- **Export Folder Paths:** Whether the export should include folder paths.
- **Export Password:** The secret whose password value will be used to password-protect the export archive.
- **Export TOTP Settings:** Whether the export should include Time-based One-Time Password (TOTP) settings.
- **Folder:** The folder the exported secrets are in. If no folder is selected, all secrets are exported. If Export Child Folders is enabled, the folder's subfolder's secrets are included.

Export Storage

Once the XML export is encrypted and archived in a ZIP file, the file is stored at an external location, which differs for on-premises customers and cloud customers. For cloud, only the 10 most recent exports are retained, and older export archives are purged as new exports are stored. This doesn't apply to on-premises customers as their local storage is used.

On-Premises storage export archives are saved in the directory in the Export Path configuration. Secret Server must have write permissions to this folder.

Cloud export archives are listed for viewing and downloading on the Automatic Export Storage tab. You can automate downloading these export archives using the REST API Automatic Export. See the "REST API Reference Download" on page 1500.

Security

Because this feature moves sensitive data outside of Secret Server, it is very important to understand that **anyone with access to the export archive *and* the export password has access to all exported secrets**. This bypasses Secret Server security features and may result in a user having access to secrets they would not have access to in Secret Server.

The export archive is a password protected, 256-bit AES-encrypted zip file. Thus, the only thing preventing a possible breach is the password, so it is important you use a cryptographically strong password to foil brute force attacks.

Permissions

The following permissions relate to automatic secret export:

- **Administer Automatic Export**: The user can do everything the other permissions allow *and* edit the automatic export configuration.
- **Download Automatic Export**: The user can view all of the automatic export tabs *and* download exports from cloud storage (cloud customers only).
- **Run Automatic Export:** The user can view all of the automatic export tabs *and* run the export manually by clicking the Run Export button.
- **View Automatic Export:** The user can view all of the automatic export tabs.

Event Subscriptions

The following automatic export events are available for event subscriptions:

- **Download:** When an export archive is downloaded (cloud customers only).
- **Edit:** When changes are made to the automatic export configuration settings.
- **Export:** When an automatic export executes, automatically or manually.

  This event also triggers when a fatal error occurs during the export causing the export to fail. Non-fatal errors do not trigger this event, but they are logged on the Automatic Export Log tab.

- **Run Export:** When an automatic export is executed manually by a user clicking the Run Export button.

  If you subscribe to both this event and the Export event, both events trigger at once.

Setting up Automatic Exports

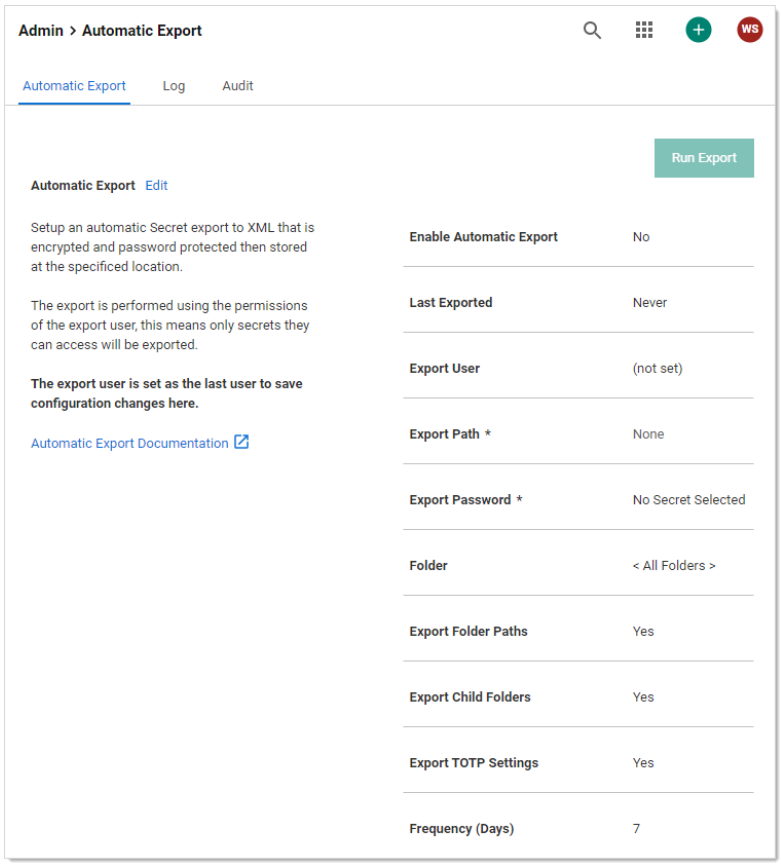
To set up an automatic export:

Only the secrets the user has view access to are exported.

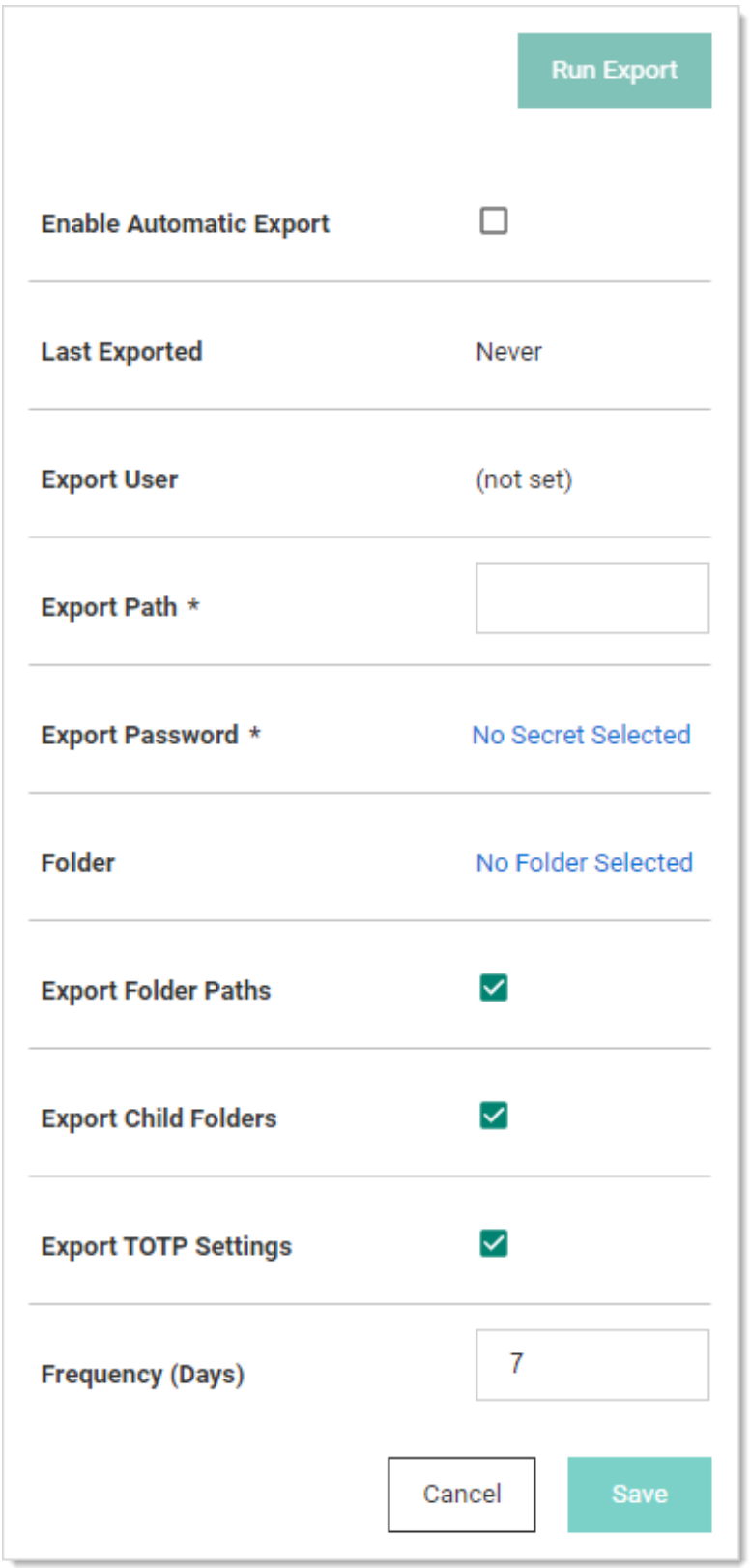
File attachments on the original secret are not exported into the XML file and require using the API to migrate. Secret audits and history are not preserved during the migration.

If you later use the export to import into another Secret Server instance, be sure to first create the AD groups and users using the permissions on the secrets in the original Secret Server instance. Otherwise, they will not be created when the secrets are imported into the new instance.

1. Create a new secret to store the export password in.
2. Go to **Admin > See All**. A popup menu appears.
3. Click the **Automatic Export** link. The Automatic Export (configuration) tab appears:



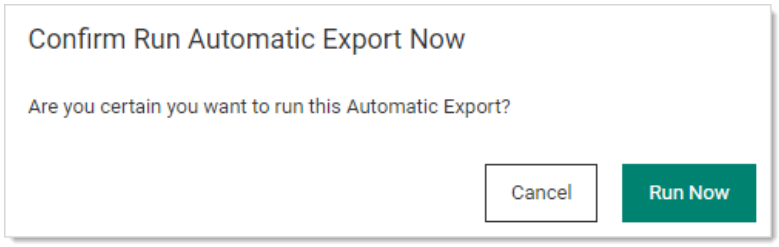
4. Click the **Edit** link next to **Automatic Export**. The page becomes editable:


Run Export
Enable Automatic Export
Last Exported
Never
Export User
(not set)
Export Path *
Export Password *
No Secret Selected
Folder
No Folder Selected
Export Folder Paths
Export Child Folders
Export TOTP Settings
Frequency (Days)
7
Cancel
Save

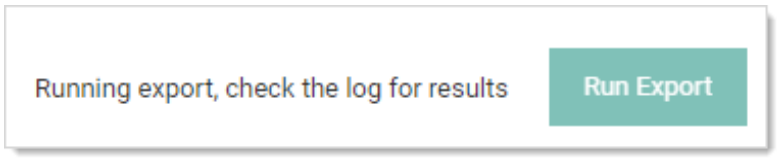
5. Click to select the **Enable Automatic Export** check box.
6. There is no need to set the export user–Secret Server automatically notes who is logged on when you save the export.
7. Type the path to export the secret to in the **Export Path** text box (on-premises users only). On-premises Secret Server customers must have write permissions to the directory.
8. Click the **No Secret Selected** link. The Select Secret popup appears.
9. Click the secret you want to store the password in. Your choice appears as a link next to Export Password:

10. Click the **No Folder Selected** link to choose a folder to export. By default, all secrets are exported if a folder is not selected. The Select Folder popup appears. Click on the desired folder. You chosen folder appears as a link:

11. (Optional) Click to select the **Export Folder Paths** check box. This adds the full folder path to the export. Folder paths in the export file provide organizational structure if secrets need to be imported later.
12. (Optional) Click to select the **Export Child Folders** check box. This option includes any subfolders of the one you chose earlier.
13. (Optional) Click to select the **Export TOTP Settings** check box if you want to include time-based one-time password settings in the export.
14. Type the number of days between automatic exports in the **Frequency (Days)** text box.
15. Click the **Save** button. Any error messages, such as secrets with doublelocks, appear. The page leaves edit mode, and the Run Export button appears.
16. Click the **Run Export** button to text the export. A confirmation popup appears:

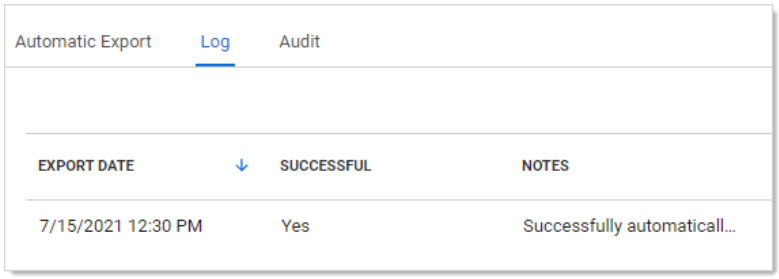


17. Click the **Run Now** button. A notification appears:



And then disappears.

18. Click the **Log** tab to confirm the export worked:



19. Using the stored password, open the zip file to confirm its contents.

Automatic Secret Export REST API

Overview

If you use Secret Server Cloud, you can use a REST API to download exports and view your export storage list.

The automatic export feature has the following endpoints available for cloud customers only. API usage is fully audited.

A typical use of these API endpoints is to automate downloading exports to your backup solution outside of Secret Server (for redundancy).

> Any permission errors when using the API will return a 403 forbidden status code and an API_AccessDenied error message.

Viewing the Storage List

Get a list of the exports currently in storage. Your session must be authenticated, and the authenticated user must have Automatic Export view permissions.

Sample Request

```
GET: http://sample.secretservercloud.com/api/v1/configuration/auto-export-storage
```

Sample Response

```
{
    "records": [
        {
            "id": 123,
            "autoExportConfigurationId": 1,
            "storageDate": "2021-07-01T07:00:02.27",
            "filename": "secret-server-export-20210707070002",
            "canDownload": true
        },
        ...
     ],
     ...
}
```

The response is a JSON object with a records property whose value is the list of all the exports in storage. Each list entry has the following properties:

- **id:** The ID for this export in storage, which is used with the Download Export endpoint to download the export.
- **autoExportConfigurationId:** The ID for the automatic export configuration this export belongs to. This may be useful in the future if we support multiple export configurations, but for now it is only used internally.
- **storageDate**: The date and time the export was stored.
- **filename:** The filename for the export archive and the export XML file inside it.
- **canDownload:** Whether the user can download this export archive.

Downloading Secret Exports

Download an export in storage by its ID. Your session must be authenticated and the authenticated user must have automatic export download permissions.

Sample Request

```
GET: http://sample.secretservercloud.com/api/v1/configuration/auto-export-storage/item/
{id}
```

Where **{id}** is the ID of the export you want to download. This value is obtained from the **Storage List** endpoint.

Sample Response

A stream of bytes representing the export archive.

Exporting Secrets

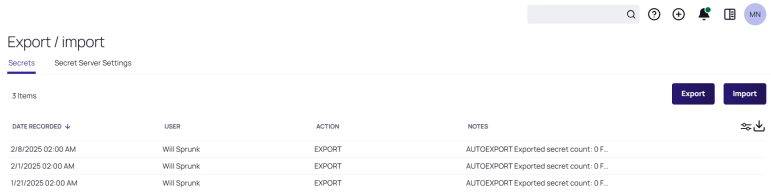
To export a secret, either CSV or XML:

Only the secrets the user has view access to are exported.

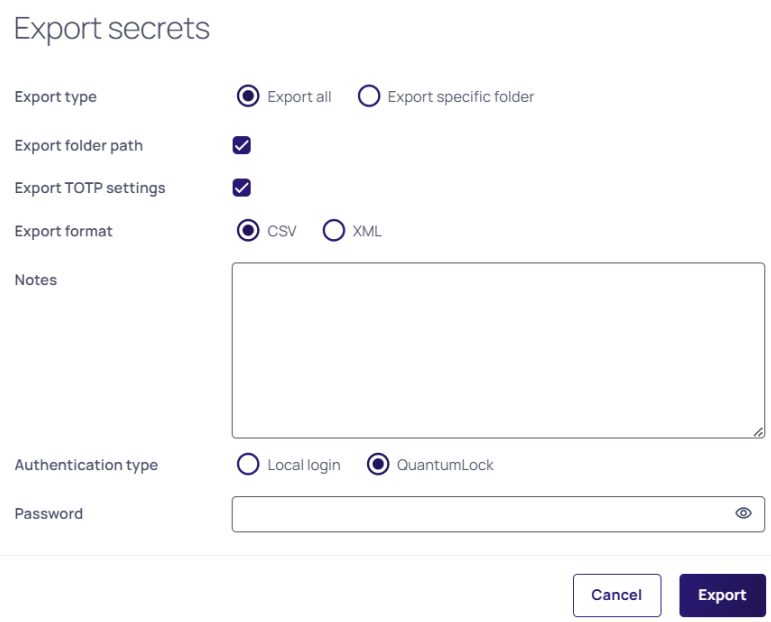
We do not support the import or export of file attachments–the API is required for that.

File attachments on the original secret are not exported into the XML or CSV file and require using the API to migrate. Secret audits and history are not preserved during the migration. Be sure to first create the AD groups and users using the permissions on the secrets in the original Secret Server instance. Otherwise, they will not be created when the secrets are imported into the new instance.

1. Go to **Admin > Export/Import** and click **Export** at the top right.



2. The Export page appears:



3. At the top, you have the option to Export All secrets or export the specific folder, all secrets are exported if a folder is not selected.
4. Select the **Authentication Type**. By default, the authentication type will be QuantumLock. The QuantumLock authentication type is required for export if you have authenticated with an external authentication service using SAML, OIDC etc.

5. Type your password in the **Password** text box. The administrative password must be entered, as it is a security measure to verify the permission of the user performing the export.
6. (Optional) Click to select the **Export with Folder Path** check box. This adds the full folder path to the export. Folder paths in the export file provide organizational structure if secrets need to be imported later.
7. (Optional) Click to select the **Export Child Folders** check box. This option includes any subfolders of the one you chose earlier.
8. (Optional) Click to select the **Import with TOTP Settings** check box if you want to include time-based one-time password settings in the export.
9. Click the **Export Format** selection button to choose the type of export. CSV is for Excel and the like, and XML is for migrating to other Secret Server instances.
10. Click the **Export** button. The Export Secrets popup appears. Any error messages, such as secrets with doublelocks, appear.
11. Click the Close button. When the exportation is finished, an `export.csv` file appears in your browser's queue:

Take care with the file—it contains unencrypted passwords.

Importing Secrets

Secret Server's importation feature simplifies integration with legacy systems and allows users to easily add large numbers of secrets from an Excel or comma-separated values (CSV) file. Secrets are batch imported by template, so multiple types of input data need to be imported in several batches. The "Secret Server Migration Tool" on page 1117 topic discusses the addition of existing passwords from other third-party password-storing applications.

We do not support the import or export of file attachments—the API is required for that.

Importing CSV Data

1. Go to **Admin > Export / Import**. The Export / Import page appears on the Secrets tab.
2. Click the **Import** button. The Import secrets page appears.
3. Select **CSV** as Import type.
4. Select the related **Secret template** from the drop down list.
5. Check to **Allow duplicate secrets** if you wish to import a secret with the same name as an existing one.
6. Check **Import with Folder** if you included an additional field in the importation text with a fully qualified folder name for the secret to be created in.
7. Check **Change remote passwords** if you wish to execute a password change for each secret on import. This enables the standard functionality of a password change, and the secret also completes the automatic

password change on checking in. This is to allow maintenance and testing of secrets protected in this manner, and a pending password change must be completed before the check-in process is allowed to begin in order to maintain a secure order of operations.

8. Check to enable **Import With TOTP Settings** if needed. If this secret has TOTP settings they will be imported, otherwise ignored.
9. Paste the secrets for importation from MS Excel or a CSV file directly into the **CSV text** text box . The order of the imported fields is based on the template selected. Consider the following:
   - Do not include a header line. The field names are determined by the order, not a header line.
   - The fields **must** be in this order: `Secret Name`, `AccessKey`, `SecretKey`, `Username`, `SecretId`, and `Trigger`.
   - Secret names must be included, but other text-entry fields can be blank unless the secret template indicates that the text-entry field is required.
   - Fields containing commas or tabs must be surrounded with double quotation marks.
   - It is permissible to include quotes. If the field is surrounded with double quotes, the double quotes you wish to include must be escaped with a \ (for example, "pa\"\"word" comes out as pa""word)
   - Values for File fields may be omitted as they are ignored by the import process.
10. Click **Preview CSV Import** - the CSVimport preview will appear below.
11. If you are happy with what you see, click **Process CSV Import**.

Importing Secrets with XML

Advanced XML importation adds folders, secret templates, and secrets based on an XML file. Permissions can be specified on the folders and secrets or the default is to inherit permissions. This import can only be done by administrators with proper role permissions.

> Migration is **not** supported by Delinea Technical Support.

Procedure

1. Ensure your XML is formatted correctly. If coming from a Secret Server export, you should be good to go. See Example XML File.

   > Do not edit the XML file with Windows Notepad. Instead, use Notepad++, Visual Studio Code, or Atom to make your edits. Windows Notepad can add invisible characters that can prevent importation.

2. Go to **Admin > > Export / Import**.
3. Click the **Import** button. The Import secrets page appears.
4. Select **XML** as Import type.
5. Check **Inherit folder permissions** to import a secret with the same folder permissions.

6. Check **Change remote passwords** if needed. The passwords on the remote device will be queued for the immediate change after import.
7. Click the **XML file** link and select the related XML file on your device to upload.
8. Click **Upload XML file**.

Example XML File

The XML file should look like the example below, the comments are for explanation only and may be removed before importing, if desired.

> Migration is **not** supported by Delinea Technical Support.

Notes

- Leaving the `<Permissions>` tag empty for a folder will cause that folder to inherit permissions from its parent folder.
- Leaving the `<Permissions>` tag empty for a secret will cause it to inherit permissions from its folder.
- To add a line-break within a Notes field use `##BR##`.

> Please do **not** edit the XML file with Windows Notepad. Use Notepad++, Visual Studio Code, or Atom to make your edits. Using Notepad increases you chances of importation failure.

Sample XML

```
<?xml version="1.0" encoding="utf-16"?>
<ImportFile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Folders>
    <Folder>
      <FolderName>Customers</FolderName>
      <FolderPath>Customers</FolderPath>
      <Permissions>
        <Permission>
          <View>true</View>
          <Edit>true</Edit>
          <Owner>true</Owner>
          <UserName>admin</UserName>
        </Permission>
        <Permission>
          <View>true</View>
          <Edit>false</Edit>
          <Owner>false</Owner>
          <GroupName>Auditors</GroupName>
        </Permission>
      </Permissions>
    </Folder>
    <Folder>
```

```
      <FolderName>Customer A</FolderName>
      <FolderPath>Customers\Customer A</FolderPath>
      <Permissions />
    </Folder>
  </Folders>
  <Groups>
    <Group>
      <GroupName>Other Administrators</GroupName>
      <GroupMembers>
        <GroupMember>
          <UserName>admin2</UserName>
        </GroupMember>
        <GroupMember>
          <UserName>DomainAdmin</UserName>
          <Domain>http://testdomain.test.com</Domain>
        </GroupMember>
      </GroupMembers>
    </Group>
    <Group>
      <GroupName>Domain Administrators</GroupName>
      <Domain>http://testdomain.test.com</Domain>
      <GroupMembers>
        <GroupMember>
          <UserName>DomainAdmin</UserName>
          <Domain>http://testdomain.test.com</Domain>
        </GroupMember>
      </GroupMembers>
    </Group>
  </Groups>
  <SecretTemplates>
    <secrettype>
      <name>Windows Account</name>
      <active>true</active>
      <fields>
        <field isexpirationfield="false">
          <name>Resource URL</name>
          <mustencrypt>false</mustencrypt>
          <isurl>false</isurl>
          <ispassword>false</ispassword>
          <isnotes>false</isnotes>
          <isfile>false</isfile>
          <passwordlength>0</passwordlength>
           <historylength>0</historylength>
          <isindexable>false</isindexable>
        </field>
        <field isexpirationfield="false">
          <name>Username</name>
          <mustencrypt>false</mustencrypt>
          <isurl>false</isurl>
          <ispassword>false</ispassword>
          <isnotes>false</isnotes>
          <isfile>false</isfile>
          <passwordlength>0</passwordlength>
```

```
            <historylength>0</historylength>
            <isindexable>false</isindexable>
          </field>
          <field isexpirationfield="false">
            <name>Password</name>
            <mustencrypt>true</mustencrypt>
            <isurl>false</isurl>
            <ispassword>true</ispassword>
            <isnotes>false</isnotes>
            <isfile>false</isfile>
            <passwordlength>12</passwordlength>
            <historylength>2147483647</historylength>
            <isindexable>false</isindexable>
          </field>
          <field isexpirationfield="false">
            <name>Notes</name>
            <mustencrypt>false</mustencrypt>
            <isurl>false</isurl>
            <ispassword>false</ispassword>
            <isnotes>true</isnotes>
            <isfile>false</isfile>
            <passwordlength>0</passwordlength>
            <historylength>0</historylength>
            <isindexable>true</isindexable>
          </field>
        </fields>
        <expirationdays>0</expirationdays>
      </secrettype>
    </SecretTemplates>
    <Secrets>
      <Secret>
        <SecretName>Test Secret</SecretName>
        <SecretTemplateName>Windows Account</SecretTemplateName>
        <FolderPath>Customers\Customer A</FolderPath>
        <Permissions>
          <Permission>
            <View>true</View>
            <Edit>true</Edit>
            <Owner>false</Owner>
            <GroupName>IT Admins</GroupName>
          </Permission>
          <Permission>
            <View>true</View>
            <Edit>true</Edit>
            <Owner>true</Owner>
            <UserName>admin</UserName>
          </Permission>
        </Permissions>
        <SecretItems>
          <SecretItem>
            <FieldName>Resource URL</FieldName>
            <Value>10.10.0.25</Value>
          </SecretItem>
```

```
        <SecretItem>
          <FieldName>Username</FieldName>
           <Value>Administrator</Value>
        </SecretItem>
        <SecretItem>
          <FieldName>Password</FieldName>
          <Value>D*KGY#$5</Value>
        </SecretItem>
        <SecretItem>
          <FieldName>Notes</FieldName>
          <Value>Just some notes##BR##...and some more notes on a new line. </Value>
        </SecretItem>
      </SecretItems>
    </Secret>
    <Secret>
      <SecretName>Another Test Secret</SecretName>
      <SecretTemplateName>Windows Account</SecretTemplateName>
      <FolderPath>Customers\Customer A</FolderPath>
      <Permissions />
      <SecretItems>
        <SecretItem>
          <FieldName>Resource URL</FieldName>
          <Value>10.10.0.25</Value>
        </SecretItem>
        <SecretItem>
          <FieldName>Username</FieldName>
          <Value>JSmith</Value>
        </SecretItem>
        <SecretItem>
          <FieldName>Password</FieldName>
          <Value>DKud3()DS</Value>
        </SecretItem>
         <SecretItem>
          <FieldName>Notes</FieldName>
          <Value>This line has an empty line##BR####BR##in between this line.</Value>
        </SecretItem>
      </SecretItems>
      <SecretDependencies>
        <SecretDependency>
          <Active>true</Active>
          <Restart>true</Restart>
          <Description>Some Dependency</Description>
          <MachineName>192.168.99.1</MachineName>
          <DependencyName>Some Service</DependencyName>
          <Type>Windows Service</Type>
          <PrivilegedAccount>Some Account</PrivilegedAccount>
          <WaitBeforeSeconds>10</WaitBeforeSeconds>
        </SecretDependency>
      </SecretDependencies>
    </Secret>
  </Secrets>
</ImportFile>
```

Secret Server Migration Tool

Secret Server offers a migration utility for importing secrets from other applications. Currently, the migration tool supports to following applications:

- KeePass (version 1 and 2)
- Password Corral
- Password Safe
- Passwords MAX

Note: This is done with another exportation tool that creates a single XML file. Please contact Delinea Support for details.

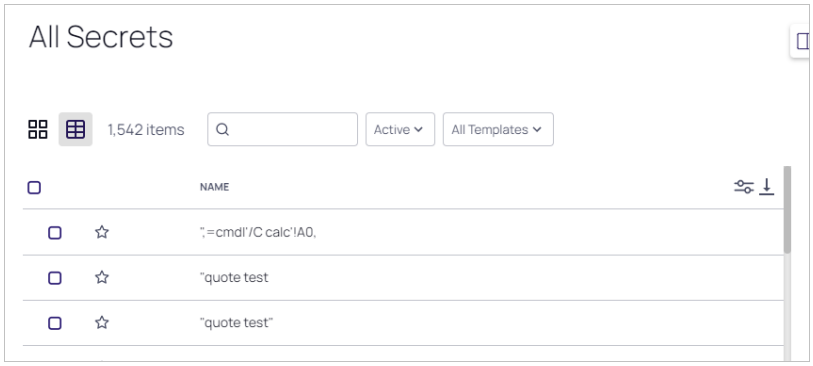
Download the Tool

Secret Management Overview

Secrets are individually named sets of sensitive information. Secrets address a broad spectrum of secure data, each type represented and created by a *secret template*. You can centrally manage secret security through sharing settings for each secret. Additionally, using folder structure, you can allow one or more secrets to inherit permissions from their parent folder. All secret text-entry field information is securely encrypted before being stored in the database, including a detailed audit trail for access and history.

All Secrets Page

All Secrets is a master table of the secrets stored on Secret Server. It is a one-stop, searchable location for examining the status and properties of secrets. It is a supplement to, not a replacement for, the "Folders" on page 1059. It lists and you can sort by secret template, heartbeat status, sync status, machine, access date, username, and much more.



Click the **Secrets** menu item in the left menu to see the All Secrets Page. Click the **>>** icon to see the All Secrets folder tree.

Secret Columns

You can customize which columns are displayed by clicking the  on the right side of the title bar. The sortable columns available are (the ones displayed by default are bolded):

- Auto Change Enabled
- Checked Out
- Checkout Enabled
- Created
- Days until Expiration
- Deleted
- DoubleLock Enabled
- **Folder**
- **Heartbeat**
- Hide Password
- Inherits Permissions
- **Last Accessed (When the logged in user last accessed a secret, not the last time a secret was accessed by anyone)**
- Machine
- **Name**
- Notes
- **Out of Sync**
- Requires Approval
- Requires Comment
- **Secret Template**
- Username

Quick Launch and Actions

You can quickly access several secret actions from the All Secrets table by hovering the mouse pointer over the secret's row:

- Quick Launch: Secrets that display a rocket icon have secret launchers associated with them. Click the icon and a Select Launcher page appears. Click on the desired launcher from the list.
- More Actions: Click the **...** icon to
  - Audit: Brings you to the Audit tab of the secret's page.
  - Details: Brings you to the General tab of the secret's page.
  - Enter Comment: Brings you to a page for entering a comment before you are allowed to view the secret. Comment ask for information such as reason for viewing and ticketing system.
  - Share: Brings you to the Sharing tab of the secret's page.

Other Features

The All Secrets page also allows you to:

- Select multiple secrets for "Running Dashboard Bulk Actions" on page 160 by clicking the secret's check box. You can also download multiple secrets in a CSV file by clicking the download icon at the top of the table.
- Select the secret as a favorite by clicking the star icon.
- View supplemental information about the secret by clicking any blank area on the secret's row.
- Click the secret's name to go to that secret's page.
- Filter the secrets shown in the table by template, search text, or activity status.
- Click the "Secret Navigation Slideout" on page 1153.

Azure Key Vault Integration

In this topic:

Introduction

Azure Key Vault (AKV) Integration simplifies management and governance of NHI's and secrets from the CSP's native vaults. With AKV integration you can centrally manage and update secrets to one or more Azure Key Vaults and rotate passwords or values more frequently. With fine grained roles and permissions, audit and logging, AKV integration provides increased governance, visibility, and awareness of secrets managed in Azure Key Vault without affecting development velocity or processes. AKV integration is available on Secret Server Cloud, the Delinea Platform, and Secret Server On Premises.

With Azure Key Vault Connector, you can:

- Link external vaults to Secret Server.
- Identify and categorize non-human identities into folders.
- Manage and sync secrets from external vaults to a central Delinea vault.
- Control access by applying fine-grained permissions.
- Regularly rotate secrets to maintain a strong security posture.
- Use Secret Server to keep external vaults in sync.

What is Azure Key Vault?

Azure Key Vault (AKV) is a secure cloud service provided by Microsoft Azure for storing and managing sensitive information such as secrets, encryption keys, and certificates. It offers two service tiers: Standard, which uses

software encryption, and premium, which includes hardware security module (HSM) protection. The Delinea AKV connecter is a connection to AKV with GUI elements on the Secret Server side.

Key features of AKV include:

- Access Control: Uses Azure role-based access control (RBAC) for the management plane and either RBAC or key vault access policies for the data plane.
- Auditing and Monitoring: Provides logging capabilities for all key vault operations.
- Certificate Management: Enables easy provisioning, management, and deployment of TLS/SSL certificates3.
- Encryption: All secrets are encrypted at rest using a hierarchy of encryption keys protected by FIPS 140-2 compliant modules.
- Key Management: Facilitates the creation and control of encryption keys used to protect data.
- Secrets Management: Securely stores and controls access to tokens, passwords, API keys, and other sensitive data.

AKV helps solve various security challenges in cloud environments, supporting the "use least privilege access" principle of the zero trust security strategy. It centralizes the storage of application secrets, reducing the chances of accidental leaks. To use AKV, it must be associated with a resource group within the same application/environment combination. Access to AKV is controlled through two interfaces: the management plane for managing the vault itself, and the data plane for working with the stored data.

What Is Distributed Vaulting?

Distributed vaulting is a security approach that stores and manages sensitive data, such as encryption keys, secrets, and certificates, across multiple locations, systems, or environments. This decentralized architecture provides several benefits:

- Centralized Secret Control: Store, manage, and rotate secrets from a single interface. Enforce consistent access policies and permissions across all secrets. Unified view of all secrets with a single source of truth.
- Competitive Advantage: By implementing distributed vaulting, organizations can gain a competitive advantage by enhancing security, agility, compliance, and customer trust while reducing costs and improving business continuity.
- Enhanced Availability: Data is available even if one location or system is compromised or experiences downtime.
- Improved Development Environment: Securely manage all cloud secrets without impacting developer velocity or CI/CD pipelines. CI/CD (Continuous Integration/Continuous Deployment) pipelines are automated workflows that streamline the software development process. They integrate, test, build, and deploy code changes, ensuring faster, more reliable, and higher-quality software releases.
- Improved Security: By spreading sensitive data across multiple locations, you reduce the attack surface and make it more difficult for unauthorized access.
- Increased Scalability: Distributed vaulting allows for easier expansion and adaptation to growing security needs.
- Connecting with Your Legacy Delinea Vault: Integrating Secret Server On-Premises with Delinea Platform.
- Reduced Single Point of Failure: No single location or system holds all sensitive data, minimizing the risk of catastrophic data loss.

Terminology and Concepts

AKV integration uses several new terms and concepts. Some of the term definitions are slightly different than common usage.

Auditing

All changes to linked secrets are audited and the audit grid indicates how many items we changed. Expanding the panel by clicking on the row shows the changeset that includes the changes for each update. Permission updates include what permissions were assigned or removed from which user.

Creating a Vault

"Creating a vault" links an existing external vault to Secret Server. You are *not* creating the actual external vault. That is, you are creating its internal representation within Secret Server with the external vault. The name must exactly match the name of an already existing external vault. The credential secret should have Get, List, and Set permissions within Azure under Secret Management Operations.

New Vault Initial State

After successfully validating the connection to the external vault you are prompted to pull in the matching information from the vault. This process only pulls in links to the existing external secrets inside Secret Server. At this point, no data is updated in the external vault.

Secrets first appear as disabled. A disabled secret means Secret Server will not push or pull any data to or from the external vault for that secret.

External Secret

An *external secret* is a secret inside Secret Server that is linked with a secret in an external vault. It is called an *external* secret because it *represents* a linked secret in the other vault.

In short, an external secret is mostly just a metadata mapping to a secret in the external vault.

External Secret Fields

An external secret contains the following fields, which are available on the External Secret page:

- External Vault: The vault on the external machine that contains its matching secret.
- Name: The name of the secret, which cannot be changed.
- Last Push: Indicates the last time a change was pushed to the linked secret on the external vault.
- Linked Secret: A secret in Secret Server that is connected to the external secret and thus to a secret in the external vault. Any changes to it are pushed to the external secret.
- Transform: The formula used to push changes to the linked secret on the external server. For example: `Machine: $secret.field.machine; Password $secret.field.password` would push the value of the machine and password fields into the the linked secret in the external vault. There is a formula editor that shows available fields once a secret is selected.

An external secret can have one of the following states:

- Enabled: Indicates the secret is live and any changes to it triggers an update to the external vault.
- Disabled: Indicates the secret cannot receive any changes. That is, no changes can be pushed to this secret from the external vault.

External Secret Actions

There are several actions that can be taken with an external secret:

- Set External Value: This function accepts any text and assigns it to a secret in the external vault. This function does not require a linked secret or transform and will ignore any of those and just assign the value that is entered.
- View External Value: View the current value for a secret in the external vault, not necessarily a linked secret.
- Push: Merge the transform data from the linked secret and update the value in the external vault. New versions of the external secret will only be added if it has changed values.
- Edit: Edit the secret's metadata.

External Secret Grid

The external secret grid provide a central location in Secret Server to manage external secrets. When selecting external secrets in the grid, you can select to push or edit these items. Bulk edits allow you to update and link multiple external secrets at once.

External Vault

External vault is a vault that is outside of Secret Server—one AKV is hosting. That external vault is where default permissions are assigned via the connector, and you can perform a couple of actions on that vault:

- Push: Update any active secrets in the external vault that are linked with a transform to Secret Server.
- Pull: Retrieve all the secret names in the external vault and create a pointer record.
- Synchronize: Performs a pull (from the external vault) and then a push (to the external vault). Once completed both Secret Server and the external value are updated with the other's changes.

Permissions

Permissions are assigned to the external vault and any secret within the vault uses those permissions by default. On each secret, you can override the vault permissions and assign completely different permissions.

Role Permissions

Go to **Settings > Roles > Administrator > Permissions tab** to set these permissions.

Role permissions:

- Create External Vault Links: Can setup a connection to an existing external vault. Can then assign permissions to other users.
- View External Vaults: Can access the external vault feature but cannot manipulate external vaults.

External Vault Permissions

Vault permissions govern what a Secret Server user can do with the external vault:

- Edit Vault: Can change the settings for the vault.
- Edit Vault Permissions: Can assign any permission to any user on the vault.
- Pull: Can execute a pull on the vault.
- View External Values: Can view or set a remote value on any secret within the vault. The user also needs "View Remote Value" or "Set Remote Value" on the secret.
- View Vault: Can view the vault and all information, including permissions.

Vault Secret Permissions

These permissions can be defined on the external vault as well as each secret. The values assigned on the vault are the default permissions used by any secret that inherits permissions from the vault.

> Note: When viewing a Delinea Secret, an “External secrets” tab appears that lists all of the external secrets linked to the secret.

External vault secret permissions:

- Edit External Secret: Able to change any of the fields on the secret including status, linked secret, and transform.
- Edit External Secret Permissions: Can assign any permission to the secret.
- Push: Can run the push action which will apply the linked secret to the transform and then push or update that value in the external vault.
- Set External Secret Remote Value: Can assign a free-form value directly to the external secret. Requires ‘View external values’ on the parent vault.
- View External Secret: Can view the secret and any of the associated information such as permissions and auditing.
- View External Secret Remote Value: Can retrieve and view the actual value for the secret in the external vault. Requires "View External Values" on the parent vault.

Connecting Secret Server with an AKV–App Registration Process

The following procedure makes AKVs available in Secret Server.

1. Create a new secret using the **Azure App Registration** template.
2. Go to your Azure portal.
3. Type `App registrations` in the search bar.
4. Click the **All applications** tab.
5. Click on your App you want to use if there are more than one.
6. Find and copy into the secret the following:

- Application or client ID.
- Directory or tenant ID .

7. Go to **Manage > Certificates and Secrets**.
8. Create a new client secret.
9. Copy the value that is generated and paste it into the **Client Secret** section of the secret in Secret Server.

> You must copy the value at the time of client secret creation as Azure will not allow you to go back and copy the value later. If you do not copy the value at that time, you have to create a new client secret.

Creating an External Vault Link

1. From the left navigation panel, select **Secrets > External Secrets**.
2. Click the **Create external vault link** button. The Create external vault link page appears.
3. Type the name of the Azure key vault you want to connect with in the **Name** text box. The name must **exactly** match the name of the key vault.
4. Click to select the **Enabled** check box if you want to push changes to the vault. Leave it unchecked if you do not want to push changes to it.
5. Click the **Select secret** link. A popup page appears.
6. If the table is blank, click the **All secrets** toggle button.
7. Select the credential secret you created during the app registration process. This is the secret that has proper access to query and write to AKV and will be used for all connections. The minimum permissions required for this secret in Azure under secret management operations are:
   - Get
   - List
   - Set (permissions)
8. Click the **Save** button.
9. You are prompted to synchronize the external vault. This step performs a pull on the vault and then a push on each active and linked external secret. This pulls all the secrets from the linked AKV into the Secret Server UI.
10. Synchronize the vault. The external vault summary page will shows these results:
    - Name of the vault.
    - State: enabled or disabled.
    - Last pull status.
    - Number of external secrets
    - Credential secret used
11. Select the **External Secrets** tab to view the list of external secrets.
12. Change the **Status** to **All states** to view the list of external secrets. All external secrets are initially disabled. This means no secrets are synchronized from Secret Server to the AKVs.

Managing External Secrets

1. From the left navigation panel, select **Secrets > External Secrets**. The External Secret grid appears. After you have linked at least one external vault, you will see the list of external secrets here.
2. In the grid you can:
   - Search for a specific secret.
   - View all secrets, enabled or disabled.
   - View only “enabled” or “disabled” secrets.
   - View or manage permissions.
   - View audit events.
   - View the log.
3. Select a secret in the grid. Now, you can perform a few more actions:
   - Set remote value: Sets a new value on the external secret in AKV.
   - View remote value: View the current value on the external secret in AKV.
   - Edit the secret and set properties.
   - Enable or disable synchronization for the secret.
   - Select a secret to link to. That secret serves as the master secret in Secret Server. You can link one or more external secrets to a single master secret. You can sync the same secret/vault from a single master secret to secrets in multiple key vaults.
   - Perform Remote Password Changing (RPC) on the master, which propagates to all the enabled linked external secrets.
   - Transform, which allows you to select the fields you want updated on the external secrets. Transforms are defined in the secret template for the master secret. For example, `Password (password)` links the password field from the master secret and updates all linked enabled external secrets with the password field's value.

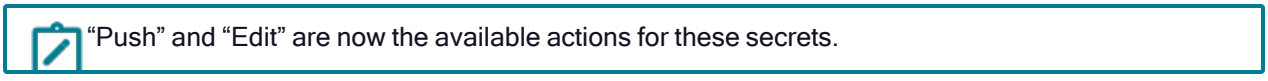
     You can also define a string format and insert field values from the linked secret using `$secret.slug.notation`. For example, `Password: $secret.slug.password` sets the external secret value as `password 1234pass` where `1234pass` is the actual password from the master secret. Transform allows you to copy and paste the slug name or simply click on the + sign to add it in the box below.

Bulk Operations with the External Secret Grid

The external secret grid provides a central location in Secret Server to manage external secrets. When selecting external secrets in the grid, you can select to push or edit these items all at once, which is a bulk operation.

This is useful for linking multiple external secrets from one or more vaults to a single master secret:

1. Select all or a few secrets from the grid.



2. Push changes from the master secret or secrets in Secret Server to the linked secrets in AKV or edit to perform additional bulk actions:
   - Toggle the enabled/disabled state
   - Select a linked secret (the master secret)
   - Perform transform actions such as updating the password or adding information
   - Select “Apply to all” or “Cancel.”

   > The heading shows the number of secrets that will be affected by this bulk action.

Simultaneously Creating Master and External Secrets in AKV

There may be times where you want to create a new master secret along with an external secret at the same time in AKV:

1. Create a secret inSecret Server using any template. For this instruction, we use the Azure AD Account template.

2. Type the basic information such as secret name, domain, username and password.
3. Click the **External Secrets** tab.
4. Click **Create** then click **Create external secret**.
5. Type the following information:
   - Name of the secret.
   - External vault where you want to create this secret.
   - Synchronization: enabled or not.
   - Linked Secret: The secret you just created.
6. If you want to sync any fields with the external secret, add those to the **Transform** section. Merge the secret field **Password (password)** to sync passwords from the master secret to the linked secret in AKV
7. Save your changes.

Secret Management Procedures

Secret management is a critical aspect of cybersecurity, involving the secure handling of digital authentication credentials, such as passwords, API keys, and encryption keys. Effective secret management ensures that sensitive information is stored, accessed, and transmitted securely, preventing unauthorized access and data breaches. Key components include creating secret policies, which are sets of rules applied to multiple secrets to enforce security settings and password requirements, and creating secrets by selecting a template and configuring settings such as automatic password changing. Customizing the All-Secrets page allows for efficient management and viewing of secrets by displaying specific columns and filtering results based on various criteria. Additionally, secrets can be deactivated, reactivated, duplicated, edited, and erased as needed to manage their lifecycle and ensure that only active secrets are in use.

Other important aspects of secret management include overriding the secret template's password requirements for individual secrets, setting up password masking to hide actual password characters, and sharing secrets with

specific permissions to facilitate collaboration while maintaining security. Viewing secrets involves accessing stored information through the All-Secrets page or individual secret pages, with permissions controlling who can view specific secrets. Secret configuration options, such as automatic password changing and expiration policies, help manage the security and lifecycle of secrets. The search functionality allows users to find specific secrets based on various criteria, ensuring accurate and up-to-date search results.

Creating Secret Policies

A secret policy is a set of rules that you can apply all at once to multiple secrets. For example, a secret policy could include rules about remote password changing or security settings, and you could apply all of the rules as a single policy to multiple secrets, whether the secrets reside in the same folder or different folders.

Follow the procedure below to create a secret policy:

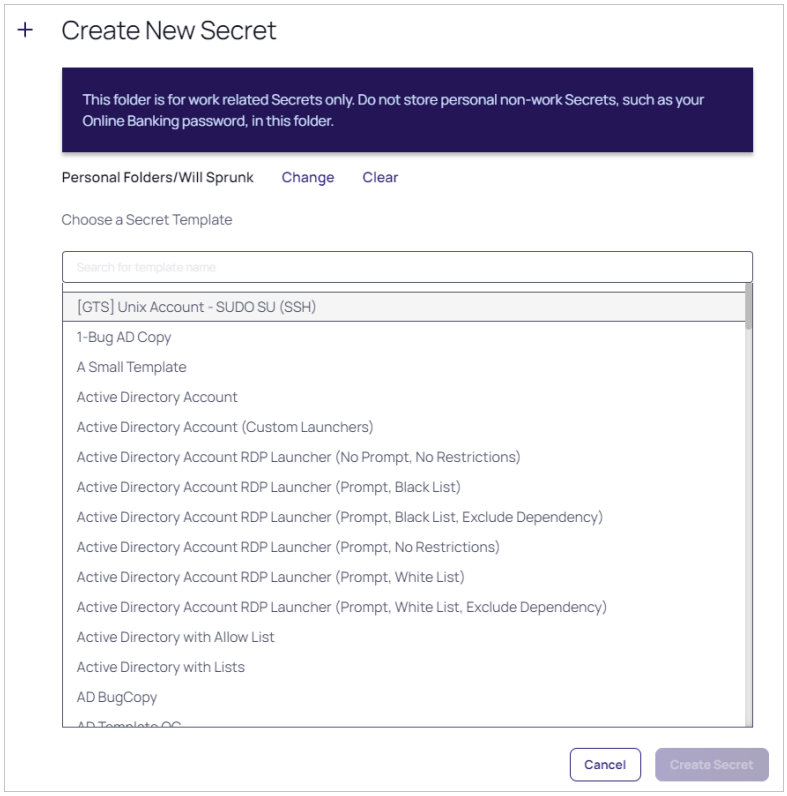
1. Navigate to **Admin > Secret Policies**. The Secret Policy page appears.
2. On the **Secret Policy** page, click **Create secret policy**. A popup appears.
3. In the Secret Policy popup type a **Name** and **Description** for your new security policy.
4. Click to select the **Enabled** check box next to the State field.
5. Click **Save**. The new policy's page appears on the Policy tab and Summary subtab.

   Because you are creating a brand-new secret policy, the value in the setting column for many policy items is *(Not Set)*.
6. Click each subtab in turn and click the **Edit** button to set which parameters are enforced. See the various topics for specifics about each parameter.

Creating Secrets

To create a secret:

1. Click the **+** icon and select **New Secret**. The Create New Secret page appears:

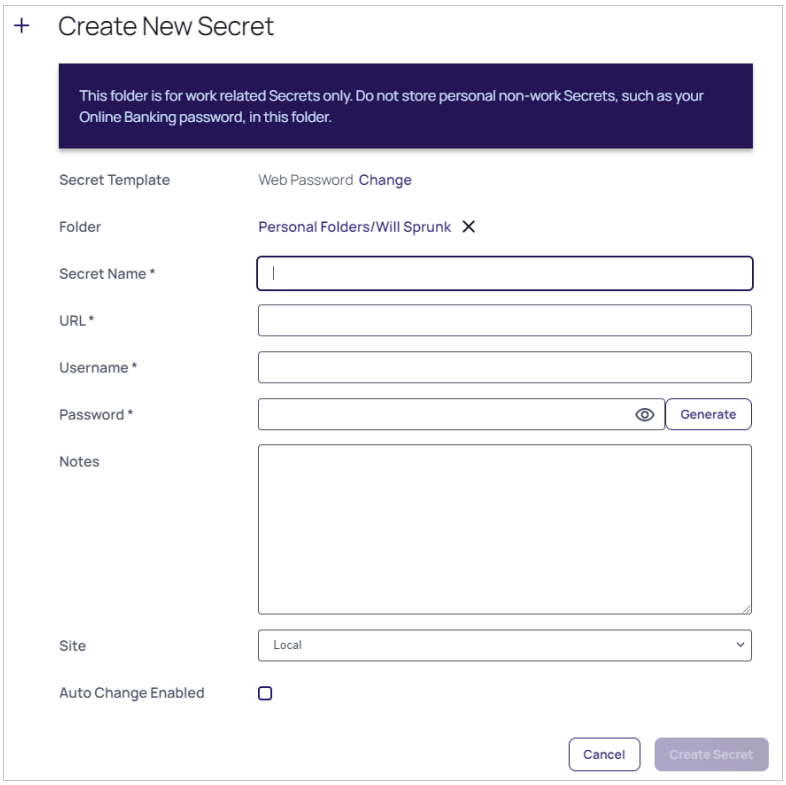


2. Click the **Choose a Secret Template** list or type a template name in the **Search for template name box** to choose a template from which to create the secret .

> If you do not find a suitable template available, you can create a custom template.

3. Click the **Create Secret** button. A Create New Secret page appears.

> These pages differ significantly, based on the secret template you chose. For this instruction, we chose the frequently used Web Password template.



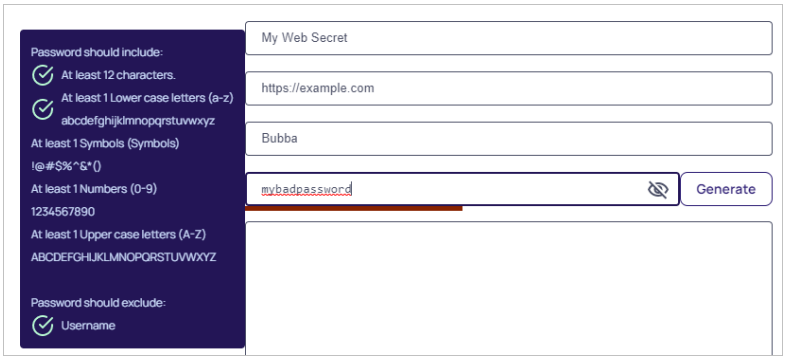
4. Complete the text boxes and selection controls on the page.

> The password generator is governed by a password requirement, which is usually set via the secret template. However, you can override the template for this secret and set the requirement to something different in the Password Requirements section of the Security tab, *after* you create the secret.

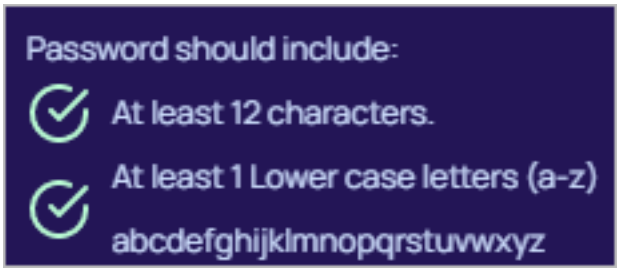
5. Click the **Generate** button to create a strong password that meets the requirements for that type of secret. You can also add your own.

> Click the eyeball icon to see a list of the current requirements and what you have already typed. The maximum password length is 1024.

The bar below the text box indicates the strength of the password you enter–it stays red but gets longer as the password quality improves.



As you meet each requirement, a green checkmark appears next to it.



Once you have met all the requirements, the color bar is full length and green.

6. Click the **Sites** dropdown list to select a site the secret belongs to.
7. (Optional) Click to select the **Auto Change Enabled** check box to enable automatic remote password changing (RPC) for the secret.
8. Click the **Create Secret** button. The new secret's page appears.
9. Click any of the tabs to further configure the secret.

> It is possible to import data as secrets. See "Importing Secrets" on page 1111.

Customizing the All-Secrets Page

On the main menu, there is a **Secrets** folder tree. When you click on the root or any subfolder, you see a list of all the secrets in that folder with multiple columns. You can customize what you see in one of three ways:

Customizing Visible Columns

You can display additional columns on the grid by clicking the icon. This data can be either secret metadata or template text-entry fields that have been set to be available for viewing. To select additional columns to display, click the **Advanced** link and then the **Column Selection** link. You can display the following metadata fields:

- Auto Change Enabled
- Checked out

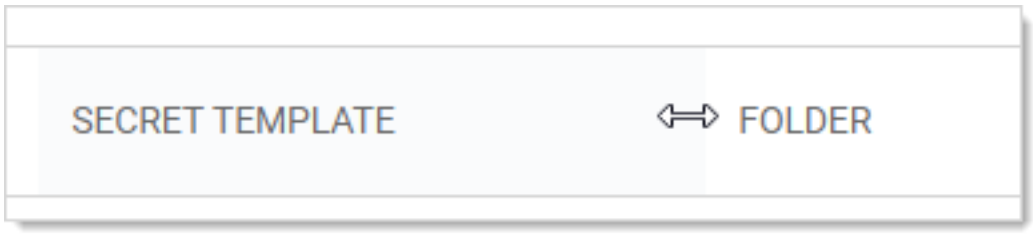
- Checkout Enabled
- Created
- Days until Expiration
- Deleted
- Double Lock Enabled
- Expiration Field Changed
- Folder
- Inherits Permissions
- Heartbeat
- Hide Password
- Last Accessed (When the logged in user last accessed a secret, not the last time a secret was accessed by anyone)
- Machine
- Notes
- Requires Approval
- Requires Comment
- Secret Template
- Username

Filtering Search Results

You can filter secret search results by selecting a folder on the left, either by clicking it or using the search text-entry field above the folder tree. On the right side of the widget, secrets can be filtered further by specifying search criteria in the top text box. The Advanced section allows filtering by secret template and status, as well as the option to include secrets contained in subfolders. Advanced criteria only remain in effect while those options are expanded (visible).

Sizing Columns

You can resize any of the columns by hovering the cursor over the border between them till it turns into a double arrow:



Click and drag to resize the column.

Deactivating and Reactivating Secrets

Secrets are **not** removed forever in Secret Server. Instead, they are *deactivated*. This maintains an audit trail for secrets, even ones that are no longer used. Administrators or users with specific permissions can view or even reactivate deactivated secrets.

In rare circumstances, such as for regulatory compliance, where deactivation is insufficient, you can instead *erase* secrets. See "Erasing Secrets" on page 1134 for details.

> We strongly recommend against deactivating large numbers of secrets, which negatively affects performance over time. Secret Server is not a transactional system–it is not designed to handle large numbers of deactivations. Deactivated secrets continue to use database table resources forever.

> Users must have ownership and the View Inactive Secrets permission to view inactive secrets. The Deactivate Secret permission is required to deactivate a secret.

To deactivate a secret:

1. Navigate to the secret **View** page by searching or drilling down the folder tree.
2. Click the **Options** dropdown list and select **Deactivate**. A confirmation appears.
3. Click the **Confirm Deactivate** button.
4. The secret is logically deleted and hidden from users who do not have a role containing the View Inactive Secrets permission.

> Secret Server uses deactivations to maintain the audit history for all data. However, deactivated secrets are still accessible by administrators (like a permanent Recycle Bin) to ensure that audit history is maintained and to support recovery. A user must have the View Inactive Secrets permission in addition to Owner permission on a secret to access the secret View page for a deactivated secret. For more information about these permissions, see "Overview of Users, Roles, User Groups, and User Teams" on page 1271 and " Sharing Secrets" on page 1144.

To reactivate a secret:

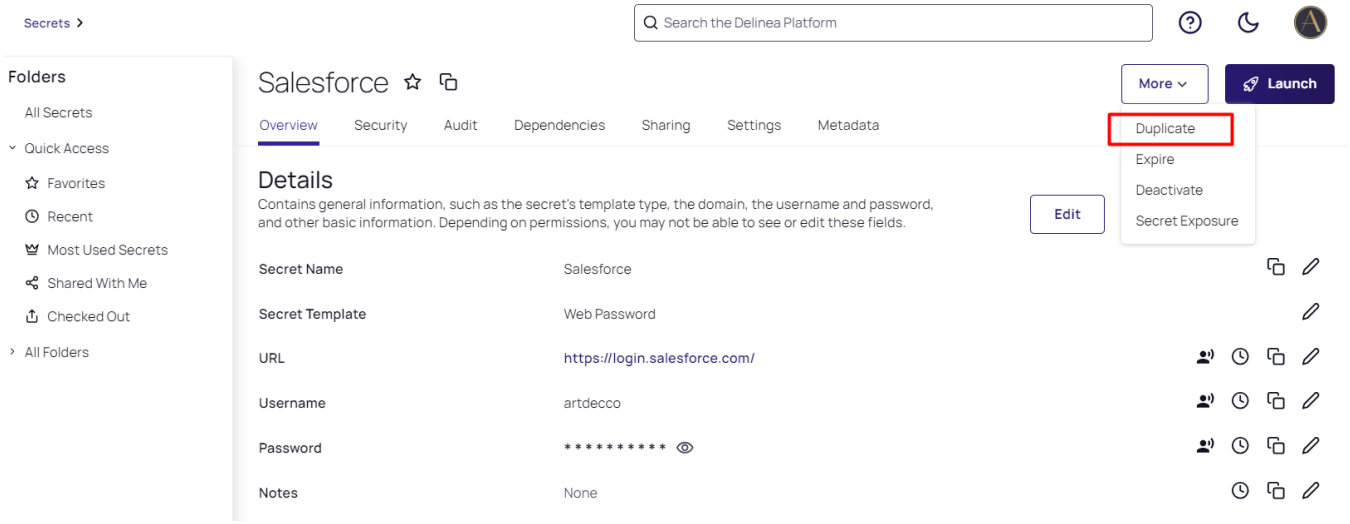
1. Navigate to the secret view page.
2. Click the **Active** menu link and select **Inactive**. The secret list now shows inactive secrets.
3. Click the name link for the desired secret. Its secret page appears.
4. Click the **Options** button and select **Activate**.

> Secrets can also be deactivated and reactivated in bulk. See "Running Dashboard Bulk Actions" on page 160.

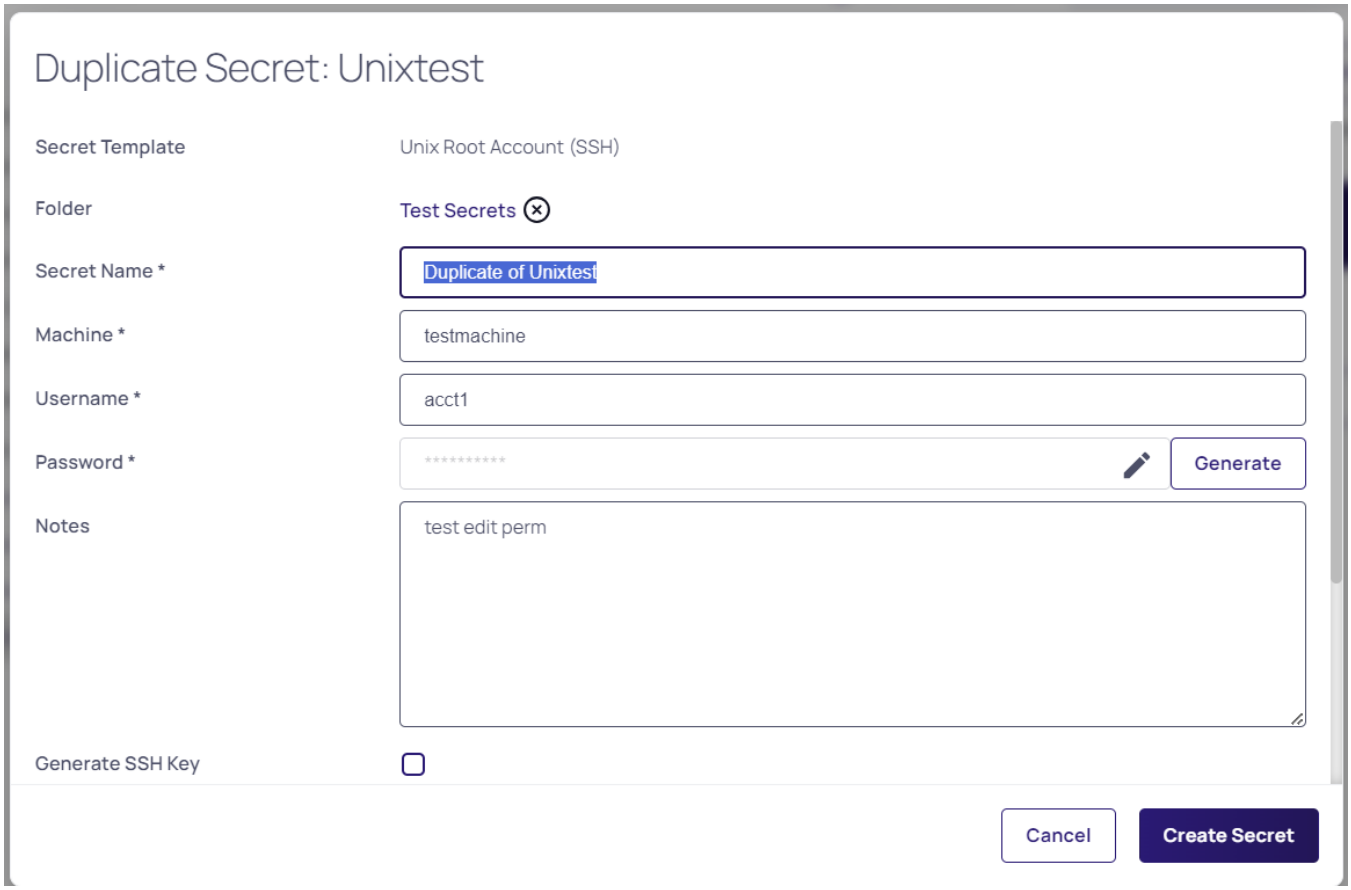
Duplicating Secrets

The secret duplication function allows for easier, automatic secret duplication. Any user with the Owner Secret permission on a secret can create a new secret with information based on the original secret.

1. To duplicate a secret, select the related secret, click **More** at the top right, and select **Duplicate** from the drop-down.



2. Enter the duplicate secret's name, the related machine, user name and password, optionally add a note, and check Generate SSH Key if you would like the SSH key to be generated for the secret. When done, click **Create Secret**.



When duplicating a secret, the following data will be copied over from the source secret to the target:

- Text-entry field information - secret name, machine, username and password.
- Launcher settings - see Secret Launchers and Protocol Settings for details.
- Secret settings - email and expiration settings.
- Double locks - enabled or disabled.
- Permissions - see Secret Permissions for details.
- Audit records - will be written to the source secret and target secret to indicate that a copy operation took place.

> Note that currently the file attachments are not copied.

Editing Secrets

> If using the Dashboard, see "Application Dashboard" on page 157.

To edit a secret:

1. Navigate to the secret's **View** page by searching or drilling down the folder tree.
2. Click the desired tab for the secret configuration.
3. Click the **EditAll Fields** link. All text-entry fields become editable.

> The password generator is governed by a password requirement, which is usually set via the secret template. However, you can override the template for this secret and set the requirement to something different in the Password Requirements section of the Security tab after you create the secret.

4. For passwords, you can create a random password with the **Generate** button (on the General tab). This generates a password according to the rules set at the template level (see secret templates for more information about password requirements).
5. Click the **Save** button.

> For more information, see "Secret Configuration Options" on page 1148.

Erasing Secrets

Erasing a secret permanently destroys (scrambles) the secret's data and makes the secret less visible to both users and admins. For users who have access to inactive secrets, however, the secret is still visible in the folder view for auditors and other secret users to confirm data destruction.

> Erasure is primarily for regulatory compliance. Deactivation is for day-to-day operations.

Erasing and deactivating secrets are not the same thing. When deactivated, secrets are **not** removed forever. This maintains an audit trail for secrets, even ones that are no longer used. Administrators or users with specific

permissions can view or even reactivate deactivated secrets. See "Deactivating and Reactivating Secrets" on page 1132 for details.

> We strongly recommend against erasing large numbers of secrets, which negatively affects performance over time. Secret Server is not designed to handle large numbers of erasures. Erased secrets continue to use database table resources forever even though their data is destroyed. Erasure is not for cleaning up the secret database.

> Ownership and the **Erase Secret** permission are required to erase a secret. In addition, the erasure must go through an approval process. Users cannot approve their own erasure.

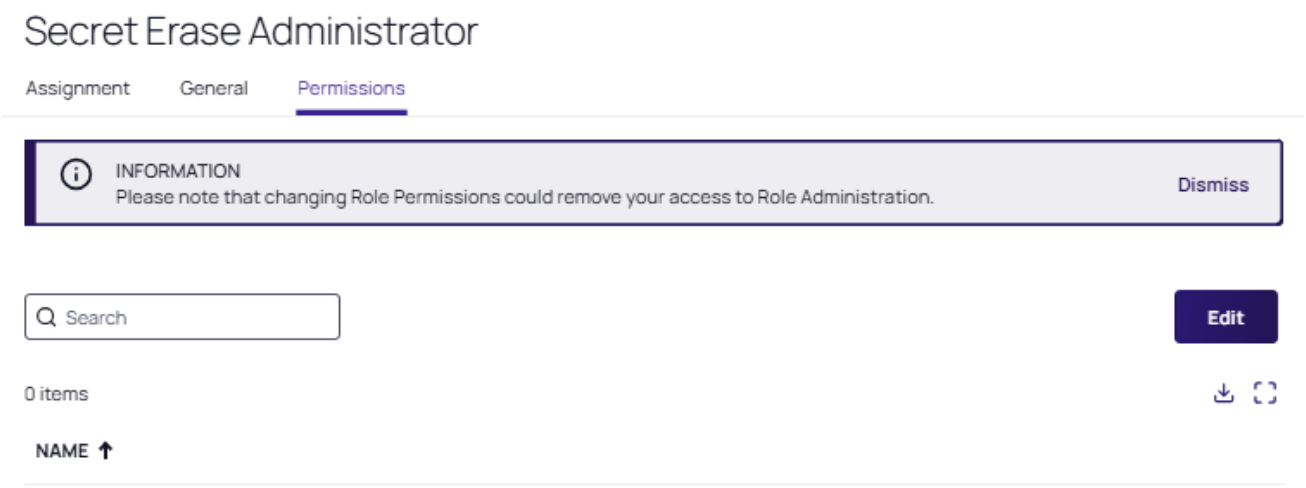
> These instructions assume you know the basics of access requests, groups, roles, and permissions. We also suggest reading the introductory material for "Workflow Overview" on page 1089 if you are not familiar with it.

Task 1: Configuring Secret Erase

> If secret erasure is already configured on this server and you are in the Secret Erasers group, you can skip to Task 2.

> These instructions are applicable for Secret Server and the Delinea Platform ununified mode or Secret Server On-Premises. If you are using Secret Server and the Delinea Platform unified mode, the new role will need to be created in the Delinea Platform

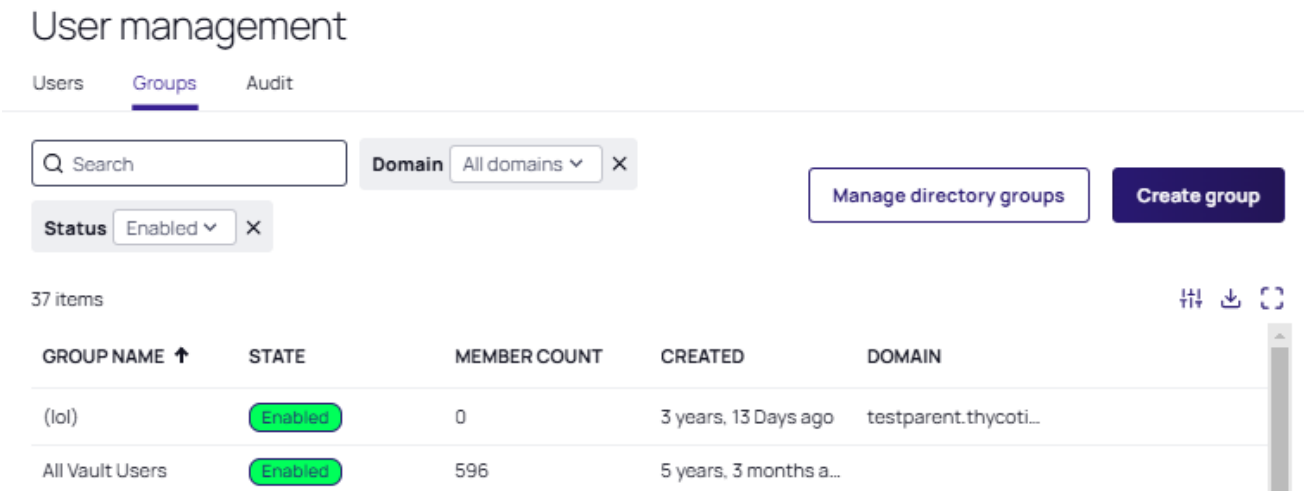
1. Ensure that you have a workflow license for Secret Server.
2. Go to **Access > Roles** in Secret Server.
3. Create a new role named "Secret Erase Requester" or "Secret Erase Administrator" (see "Creating Roles" on page 1256 for details), make sure the **Enabled** checkbox is selected.
4. Assign the **Erase Secret** permission to the secret you just created by accessing the **Permissions** tab for the role and selecting **Edit**:



In the **Scope** drop-down menu select **All**, so that all permissions available are shown, search for **Erase Secret**, select it, and click **Save**.

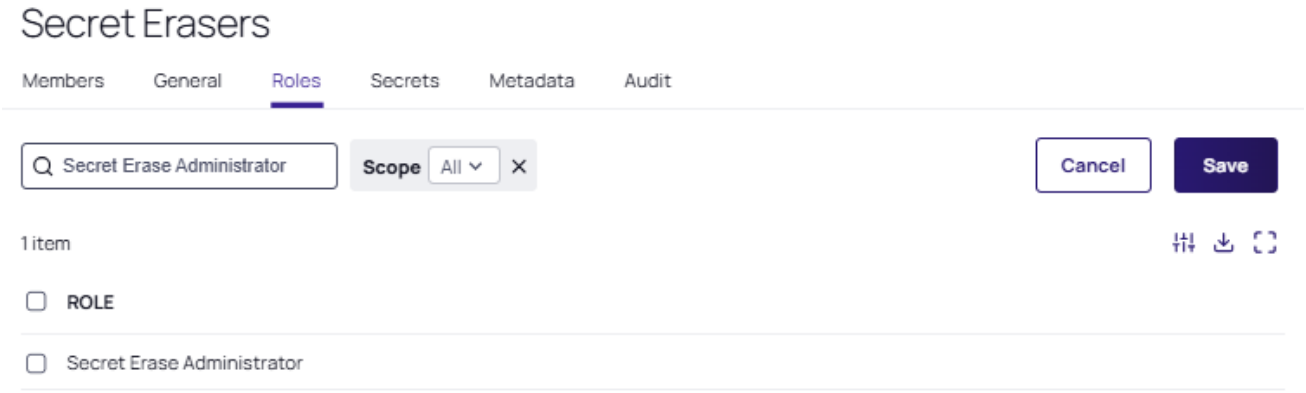
> This role permission allows users with the role to create secret erase requests and view secret erase administration pages.

5. Go to **Access > Groups**. The **Groups** tab of the **User Management** page appears:

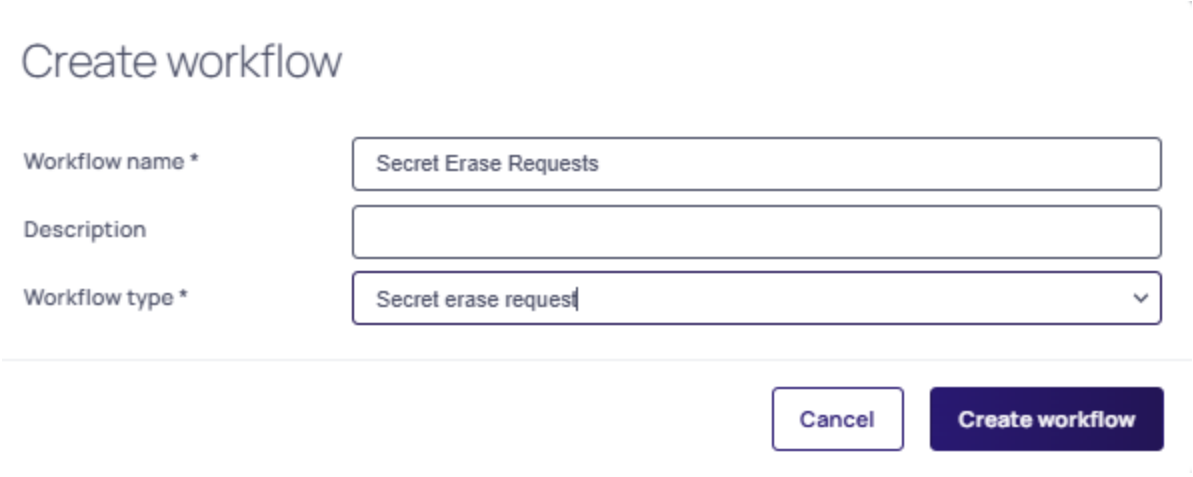


6. Create a group named "Secret Erasers", making sure the **Enabled** checkbox is selected. Once the new group is created its page loads automatically, click the **Roles** tab.

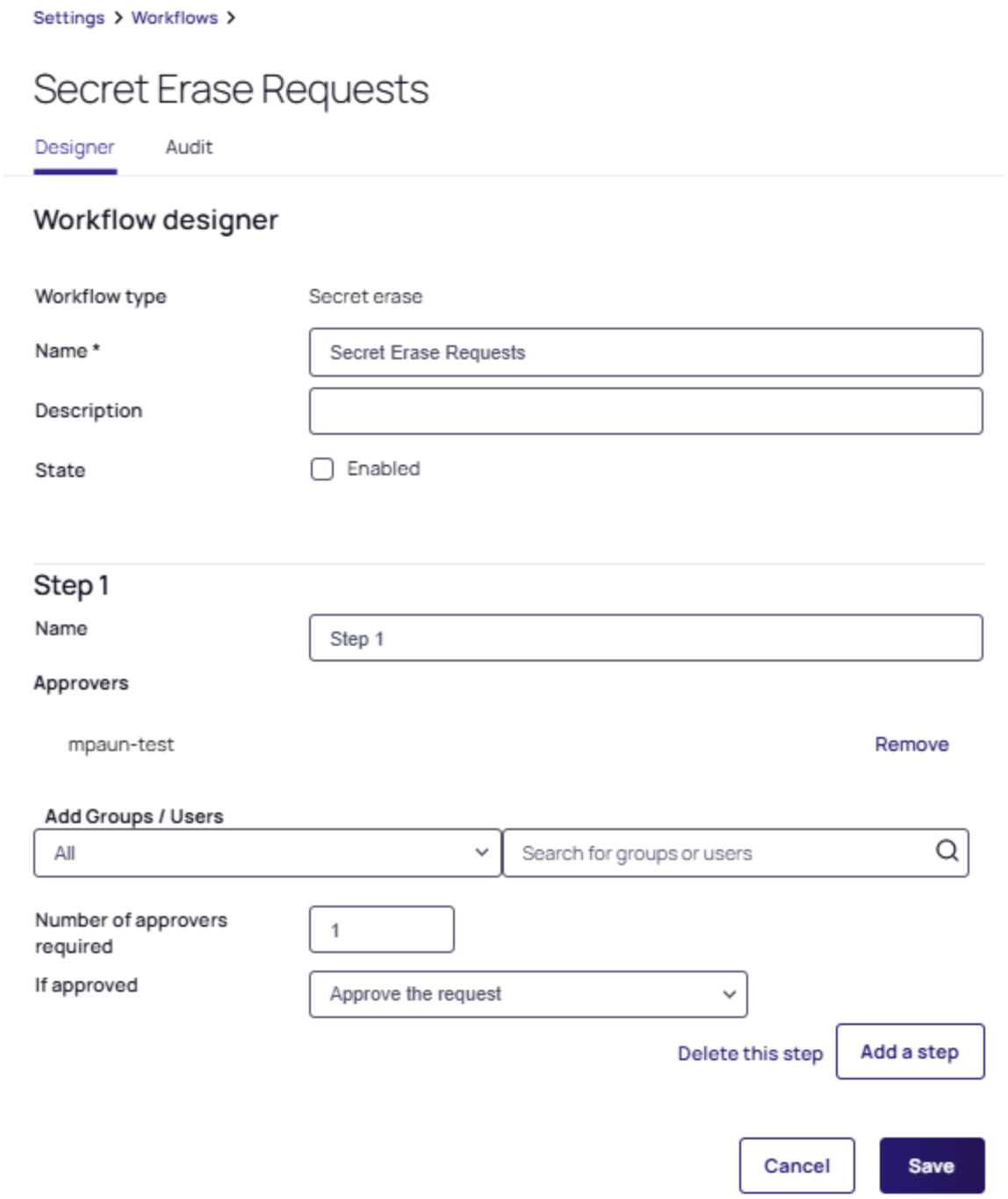
7. Click **Edit**, change the **Scope** to "All", and search for the role you just created, select it and click **Save**:



8. Click the **Members** tab to add yourself to the **Secret Erasers** group.
9. Go to **Settings > General > Workflow**.
10. Create a "Secret Erase Requests" workflow template, assigning it the **Secret Erase Request** type:



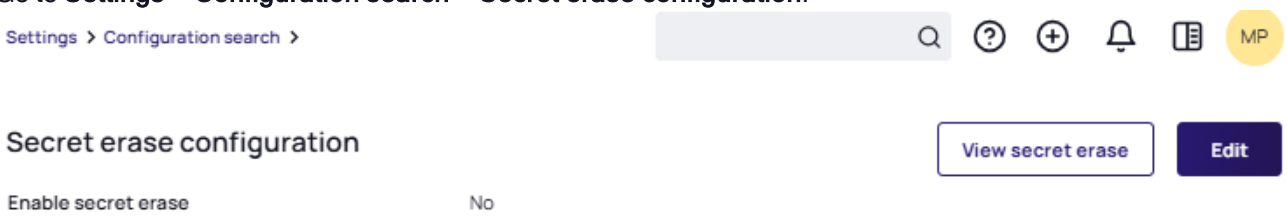
11. The workflow designer (**Designer** tab) loads automatically upon creation. In this tab, assign one or more users or groups as approvers by typing each in the search text box in the **Add Groups / Users** section and then selecting your choice when it appears. This selection then appears in the **Approvers** list box:



When satisfied, make sure the **Enabled** state is checked and click **Save**.

> Typically the approvers you choose should be in the same group as those that can make the requests. You can, however, choose any groups or users you like or make a group just for approvals. It is important to note that the same *user* cannot make both the request and approve it, in order to avoid a single person making an irreversible, potentially very harmful, mistake.

12. Go to **Settings > Configuration search > Secret erase configuration**:

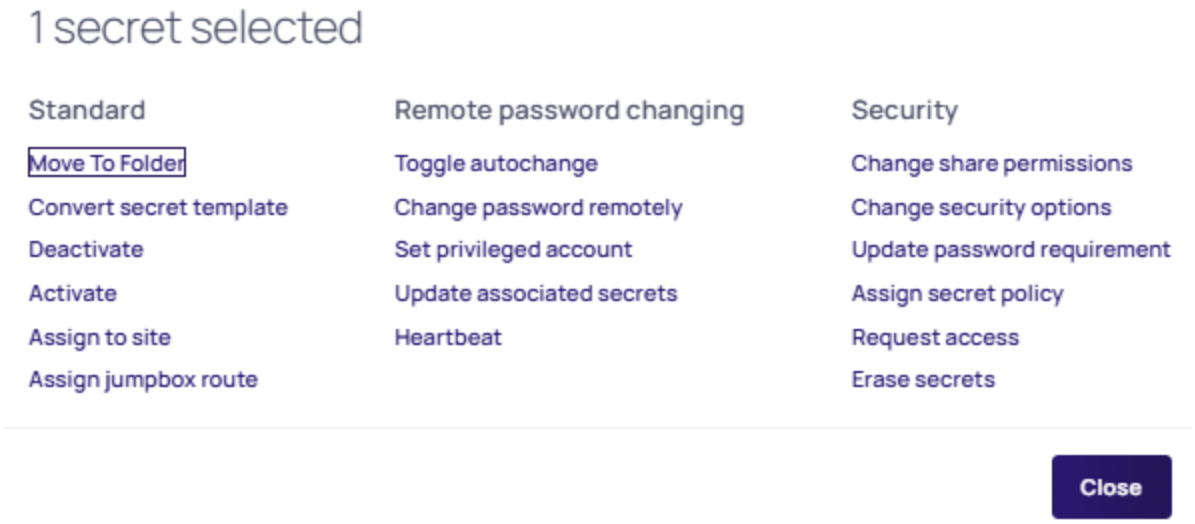


13. Click **Edit** and select the **Enable secret erase** checkbox. The **Secret erase workflow** dropdown list appears.
14. Choose from the dropdown list **Secret Erase Request** and click the **Save** button. Secret Erase is now set up.

> If the **< None >** option is the only one available in the dropdown, you will not be able to complete the setup because a valid **Secret Erase Workflow Template** is required to enable **Secret Erase**.

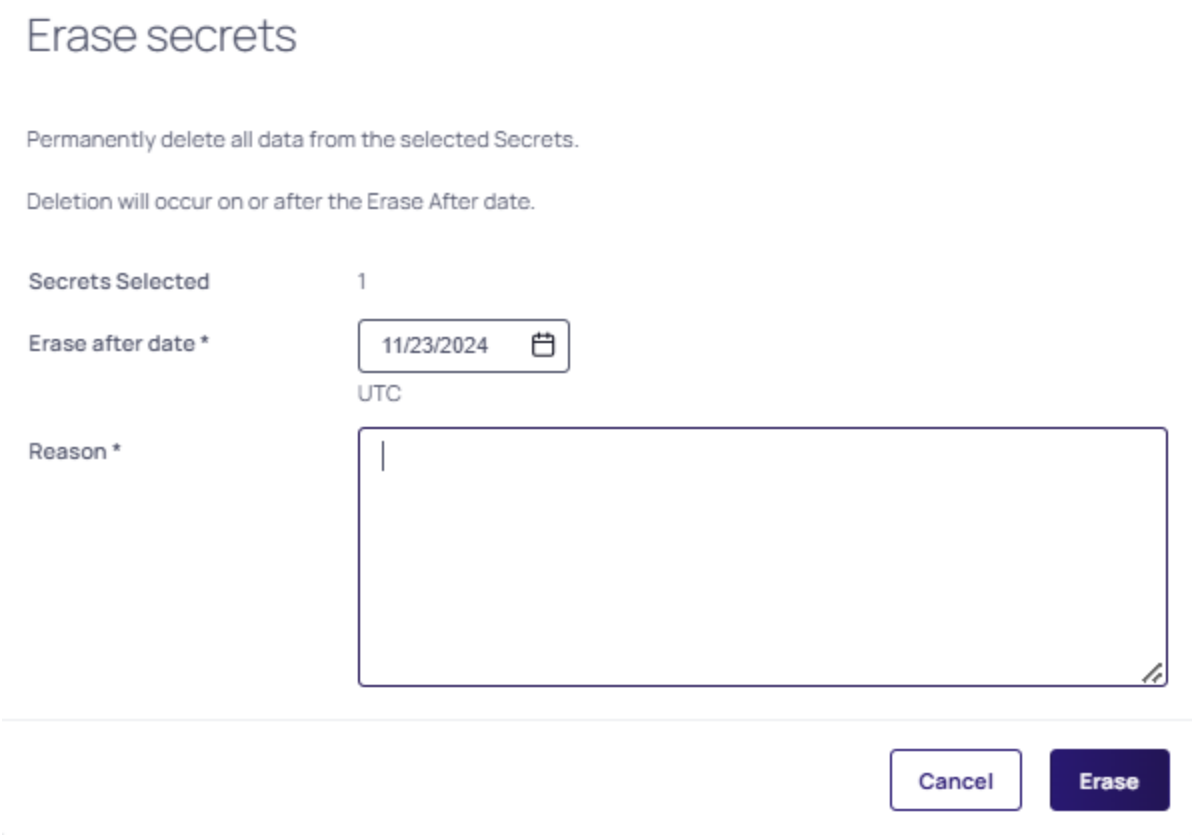
Task 2: Erasing a Secret

1. Ensure the following requirements are met for the secret you intend to erase. Ensure the secret:
   - Is inactive
   - Is owned by you
   - Does not have a pending secret erase request
   - Is not double-locked
   - Is not checked out by another user
   - Is not a discovery secret
   - Is not a domain sync secret
2. For this instruction, create a secret for testing in your personal folder to ensure all the requirements are met.
3. You can erase the secret via a dashboard bulk operation. Erase is accessed by the **Bulk Actions** button, with the **Erase secrets** option nestled under the **Security** section of the Bulk Actions popup. See "Running Dashboard Bulk Actions" on page 160 for more details:



If the "Erase Secrets" link does not appear in the **Security** section as shown above, you may not have properly configured secret erase (see Task 1), you may have not have enabled the Secret Erase Workflow or the secret might not meet one of the requirements.

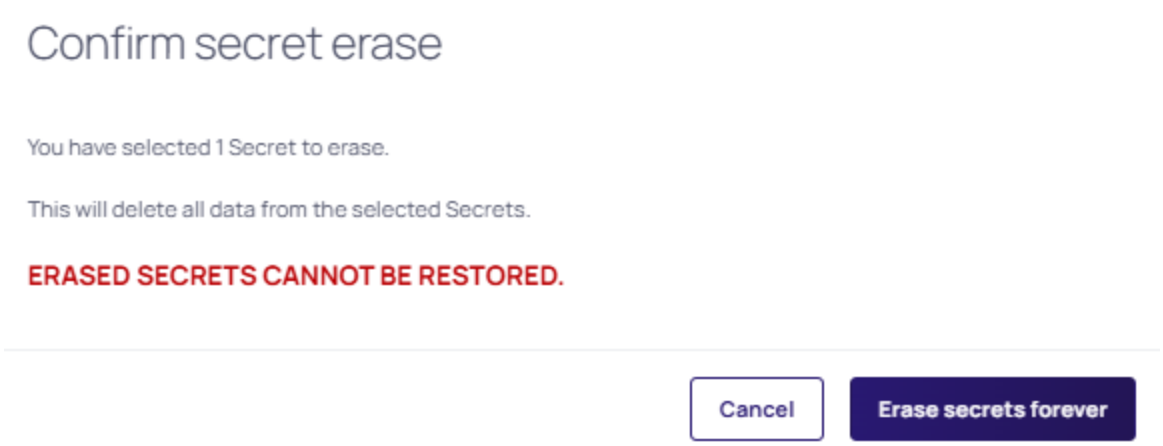
4. When you click the **Erase Secrets** link, the Erase Secrets popup appears:



   This is where you set up an erase secrets request. When you complete the process, the access request is sent to the users or user group you designated earlier for approval.

5. Use the calendar and time widgets to set the **Erase After Date**. It must be a minimum of 24 hours away to give the erase secrets request time to process. If you set it to less than that, you cannot continue the process.
6. Type your reason for permanently erasing the secret or secrets in the **Reason** text box. The granter will need this to decide whether to let you take this irreversible, destructive action. Specifically, explain why a deactivation is not sufficient. A reason is mandatory for the request to be processed

7. Click the **Erase** button. A confirmation popup appears:



8. Pause a second, to make sure you are certain.
9. Click the **Erase Secrets Forever** button.

> Errors might occur at this step if the secret is Active at the time of the request and/or if the there is no SecretId.

10. When the erase request is approved, the secret or secrets will be erased by an automated process after the "erase after" date and time arrives.

Overriding the Secret Template's Password Requirements

All secrets inherit a set of password requirements (see "Template Password Requirements" on page 1181) from their parent secret template. After you create a secret, you can choose to use a different password requirement for this one secret, which leaves other secrets based on the template as they were. To choose a different password requirement for the secret:

1. Navigate to the secret **View** page by searching or drilling down the folder tree.
2. Click the secret to open the secret's page.
3. Click the **Security** tab.
4. Click the **Edit** Link in the **Password Requirements** subsection in the **Other Security** section. The Edit Password popup appears.
5. Click the Password Requirement dropdown list to select the password requirement you desire.
6. Click the **Save** button.

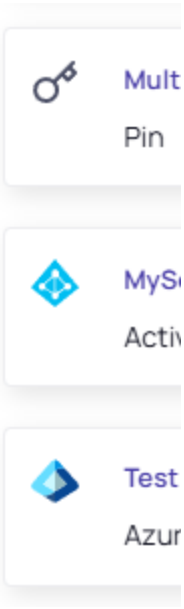
Secret Icons

Secret Icons allows you to display icons for secrets in the secret list, and secret details page.

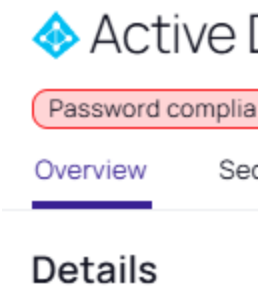
Icon Display in Different Views

Icons display differently depending on the secret view:

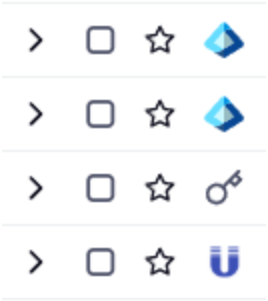
- Card View: The icon appears just to the left of the secret title.



- Detail View: The icon displays just to the left of the secret name in the title section.



- Grid View: The icon its own "Icon" column, which can be hidden or moved within the column row.



Displaying Icons

Application Level

Before you can use secret icons in any context, you must enable their use in Secret Server:

Secret icons are disabled by default.

1. Go to Settings and search for and select **User Experience**. The User Experience page appears.
2. Click the **Edit** button.
3. Click to select the **Show icons for secrets** check box.
4. Click the **Save** button.

Template Level

In the general section on any template, there is a "Display Icon" setting that allows users to select a component library icon or clear it to revert to the default skeleton-key icon.

Secret Level

In a secret's Settings tab, there is a "Display Icon" setting that allows users to select a component library icon or clear it to use the template setting icon. This supersedes any icon set on its template.

A secret's icon display varies by the presence of a URL field:

- Secrets without a URL field display the icon set on the secret. If no icon is set on the secret, the icon set on the template is used.
- Secrets with a URL field display the icon in the following order:
  1. First, the icon set on the secret is used. This is also how the URL's favicon can be overridden.
  2. If none is set for the secret. Secret Server attempts to fetch the favicon from the URL site. Favicons are icons that are representative of a website. They appear on places such as tabs and bookmarks. Secret Server's favicon is the Delinea D.
  3. If no favicon is available, the secret's template's icon is used.

Sharing Secrets

Sharing passwords is crucial for information technology teams. Due to the sensitive nature of sharing secure information, Secret Server ensures shared passwords are tracked and guarded.

Permissions

There are four permission levels to choose from when sharing secrets with another user or group:

- **View**: the user may see all secret data, such as username and password, as well as metadata, such as permissions, auditing, history, and security settings.
- **Edit**: the user may edit the secret data. This also allows users to move the secret to another folder unless the **Inherit Permissions from Folder** setting is turned on, in which case the user needs Owner permissions to move the secret.
- **List**: the user may see the secret in a list, such as a list returned by running a search, but won't be able to view any other details about the secret nor edit it.
- **Owner**: the user may change all of the secret's metadata.

Password text-entry fields are not visible if a secret has a launcher and the **Hide Launcher Password** setting is on, or if the user does not have the **View Launcher Password** role permission. See the table below for more details.

Secrets can be shared with either groups or individual users. The secret **Sharing** section allows secrets to be configured for access.

Password Visibility

Password visibility in the password text-field depends on secret access permission, role permissions, and secret security policy settings. The following table shows the possible combinations and their password visibility result.

Table: Password Visibility Determinants

Secret Access Permission	View Launcher Password Role Permission	Hide Launcher Password Policy Setting	Password Visible
Owner	No	On	**Yes**
Owner	Yes	On	**Yes**
Owner	Yes	Off	**Yes**
Owner	No	Off	**Yes**
Edit	Yes	On	**Yes**
Edit	No	On	**Yes**
Edit	Yes	Off	**Yes**
Edit	No	Off	**Yes**
View	Yes	On	No
View	No	On	No
View	No	Off	No
View	Yes	Off	**Yes**
List	Yes	On	No
List	No	On	No
List	Yes	Off	No
List	No	Off	No

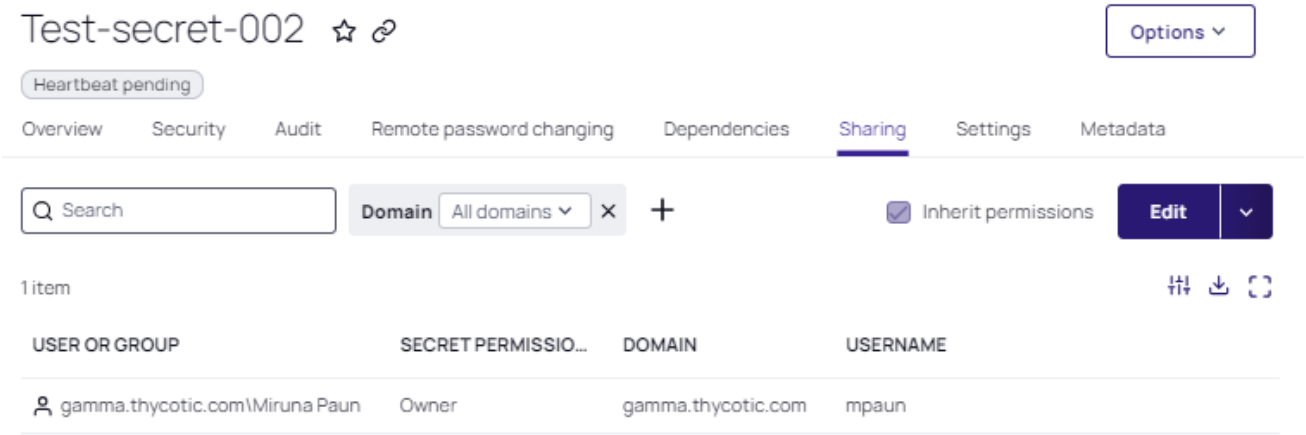
Procedure

To simplify the sharing process, new secrets automatically inherit the settings from the folder they are stored in. That is, we enable the **Inherit Permissions from Folder** check-box on the **Sharing Edit** page by default, so secrets inherit all the parent folders' sharing settings. As long as this check box is selected, you cannot set the permissions for the secret, *so you must deselect this option to add or remove users/groups you wish to share the secret with*. For more on folder security, see the "Folders" on page 1059 section.

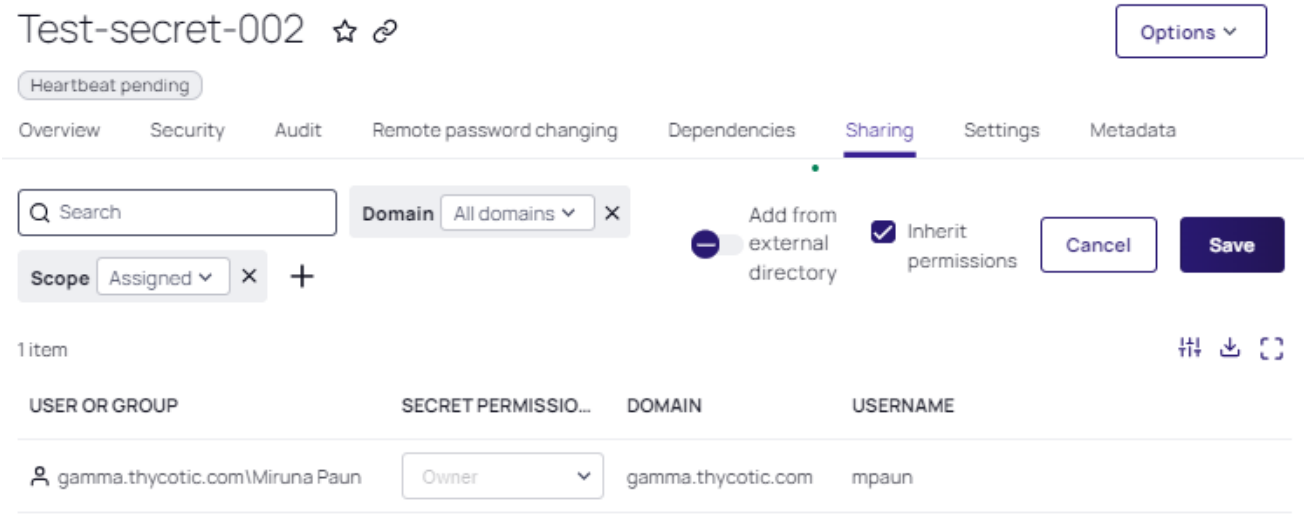
> If integrated with Delinea Platform and not restricted by teams, as well as having the **Administer Platform Integration** or the **Add From External Directory** permissions, you will see a toggle to **Add From External Directory**. Enabling that toggle will allow you to search directories connected via Delinea Platform and add new users or groups when sharing secret permissions.

To add or remove secret sharing, do the following:

1. View the secret you want to share.
2. Click the **Sharing** tab:



3. Click the **Edit** link. The page becomes editable:



   Make sure the **Inherit permissions** checkbox is deselected so that editing is possible.
4. Change the **Scope** dropdown setting to **All** to view all users and groups, assigned and unassigned alike.
5. Type into the search text box any user, group name or keyword in their title, whom you want to add to the assigned list, or alternatively remove from the assigned list for the secret.
6. When the user or group appears in the dropdown list, select it, grant it one of the four permissions, and **Save**. The user or group appears in the **Shared with** list in the **Sharing** tab automatically.
   Clicking **Cancel** will undo any changes and clicking save will apply all pending changes.
7. Repeat the process for additional users or groups.

You can also modify sharing settings for users or groups that already have sharing enabled for the secret. If a user or group is not displayed, they do not have access to the secret.

> The **Add from external directory** option searches all identity sources configured in Delinea Platform and allows creation of those users and groups in secret-server.

Troubleshooting

When a user cannot see users/groups to share secrets with, despite having been granted all necessary permissions, ensure you haven't customized their permissions outside of the built-in roles. Verify if the user is part of a Secret Server Team, and if so, they need the **Unrestricted By Teams** permission to view users/groups. See User Teams for further details.

Viewing Secrets

To view the information contained in a secret:

1. Locate the desired secret in one of these ways:
   - On the main menu, drill down the folders tree to select the secret.
   - Click the **Secret** menu item on the main menu and find the secret in the **All Secrets** table. You can filter the list or click the magnifying glass icon to search for the secret.
2. Click on the secret's name link. The secret's view page opens to the General tab.
3. Click the desired tab to view specific information. For example, click the General tab and go to the Expiration and Heartbeat section to see if the secret's password has expired and what its expiration interval is. You can check the history of the secret by clicking the Audit tab.
4. "Editing Secrets" on page 1134 if desired.

Secret Configuration Options

Common Configuration Options

These are the configuration options that are common to every secret:

- **Convert Template:** Change which template is being used to store and display information in this Secret.
- **Copy Secret:** Create a duplicate copy of the secret, which may also be renamed and modified.
- **Delete:** Delete the secret.
- **Edit:** Edit the secret parameters.
- **Favorite:** Click the star from the Dashboard or check this box on the Secret View page to mark the Secret as a favorite. It then displays in the Favorite Secrets widget.
- **Folder:** Folder location of the secret. The secret inherits permissions of this folder, depending on the Default Secret Permissions setting in the Secret Server Configuration options.
- **Share:** Configure the sharing settings, or permissions, for the secret.
- **View Audit:** View the secret audit log to see which users have accessed the secret and the actions that have been performed.
- **Site:** The site serves as the designated location for storing secrets. Any operations necessitating a Distributed Engine (DE) will utilize the site specified, along with the DEs associated with that site.

Advanced Configuration Options

These are the buttons, fields, and icons that are available for more advanced secrets:

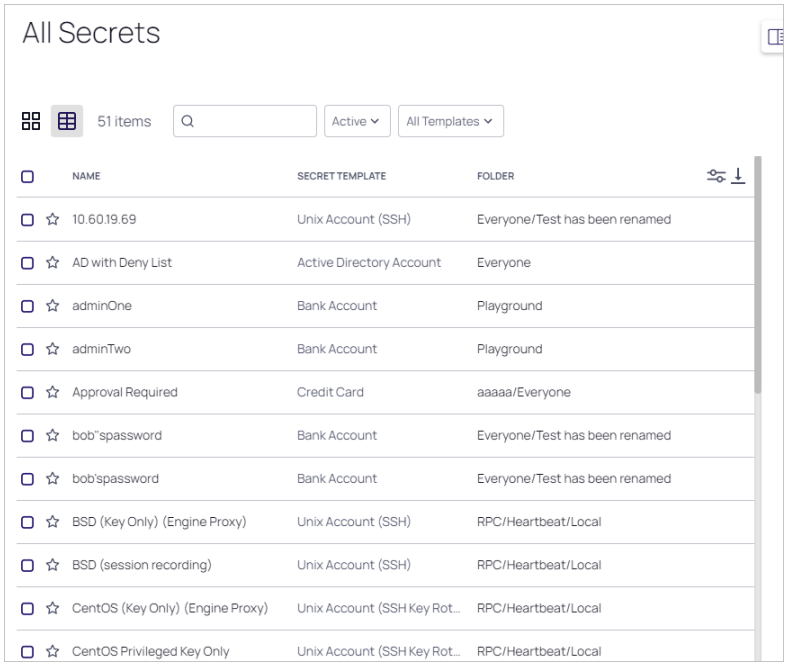
- **Expire Now:** Expire the secret manually.
- **RDP Launcher Icon:** Click to open the Remote Desktop Protocol (RDP) Launcher. See further details in the Launcher section.
- **Run Heartbeat:** Initiate heartbeat, which attempts to verify that the secret credentials can authenticate.
- **Site:** Edit the secret to set the site on the secret. This determines where password changing, heartbeat, and proxied sessions run from.

Searching and Search Indexer

Searching for Secrets

To search for secrets:

1. Click the **Secrets** menu item in the main menu. The All Secrets page appears:



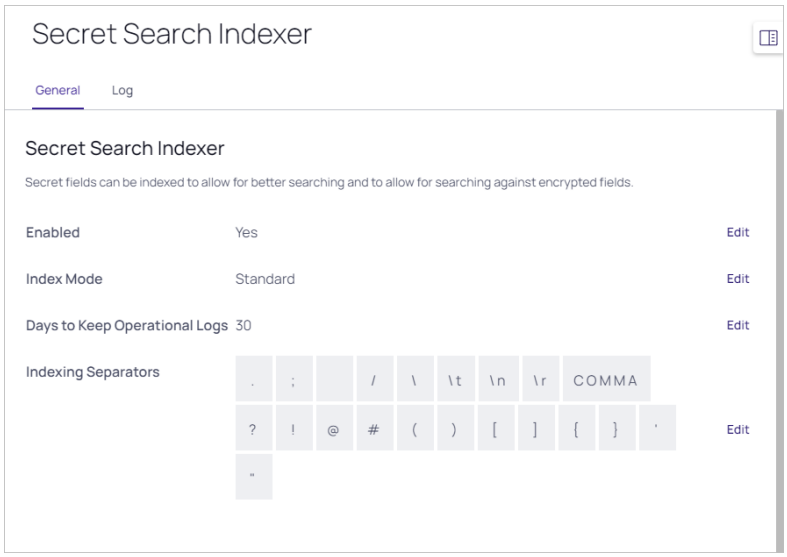
2. Type the secret name or other text in the unlabeled search text box at the top of the page.
3. Click the button. The All Secrets table only displays matching secrets. Searches search for all text-entry fields that are configured as searchable on the secret's template if the extended search indexer is enabled.

> If the search indexer is not enabled, searches are only performed on the **Secret Name** text field.

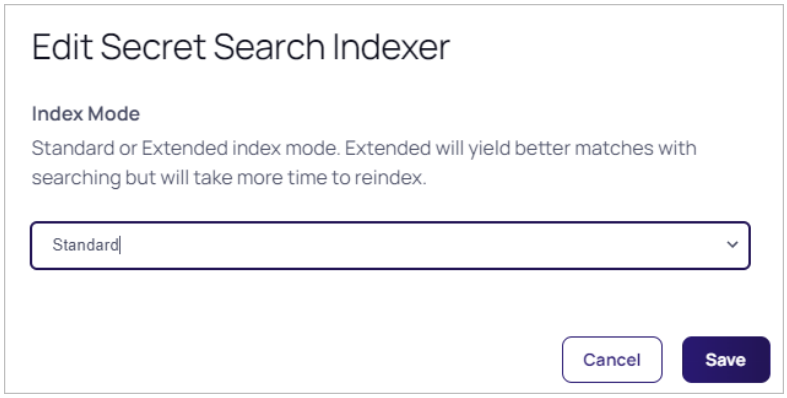
Search Indexer

The *search indexer* allows searching on all text-entry fields set to searchable on the template. To enable and configure the search indexer:

1. Click the **Admin** button on the main menu.
2. Click the **>>** icon to view the Admin slideout.
3. Type and then click Search Indexer in the Search text box. The Indexing Service page appears:



4. Click the **Edit** link next to it to ensure the **Enabled** check box is selected. Click **Save** in the popup if you changed anything.
5. Click the Edit link for the Index Mode. A popup appears:



6. Click the dropdown list to select either **Standard** or **Extended**:
   - *Standard search mode* is the default and searches on whole words in a field value. For example, a field value of "My AWS Secret" would match when you search for *My AWS Secret*, *My*, or *Secret*.
   - *Extended search mode* searches for whole words or a partial words by up to twelve characters. For example, a field value of "My AWS Secret" would match when you search for *My AWS Secret*, *My*, *Secret*, *WS*, or *ecret*. This is more useful, but may impact search performance and creates a larger index table.

   Indexing separators are used to split the text text-entry fields into search terms. By default, the separators are semi-colon, space, forward slash, back slash, tab (\t), new-line (\n), return (\r), and comma. Changes to the indexing separators require a full rebuild of the search index.

7. Click the **Save** button.

8. Click the Edit link to change the **Days to Keep Operational Logs** spinner to set the period to keep indexing-related logs that might contain PII. Secret Server automatically deletes logs older than that (in days).
9. Click the **Save** button. The Indexing Service page reappears, and the indexing begins in the background. Depending on the size of the Secret Server installation, it may take awhile. Progress is shown on the Progress bar.
10. If you changed the indexing separators, click the **Rebuild Now** link.

Secret Expiration Overview

Secret expiration is a core feature of Secret Server that enhances security by ensuring sensitive data is regularly reviewed and updated. Secret expiration allows any template to be set to expire within a fixed time interval. For a secret to expire, a specific field, such as a password, must be selected as the target of the expiration. For example, a secret template for Active Directory accounts might require a password change every 90 days. If the password remains unchanged past the specified time, the secret is considered expired and appears in the Expired Secrets panel on the Dashboard or Home page.

Benefits

- Security: Reminds users to review and update sensitive data regularly.
- Compliance: Helps meet compliance requirements that mandate regular password changes.
- Automation: When combined with RPC, Secret Server can automate the process of changing passwords regularly.

Setting Up Secret Expiration

1. Enable Expiration on the Template:
   - In the Template Designer, enable the `Expiration Enabled?` checkbox.
   - Set the number of days until expiration and select the field to be updated upon expiration.
2. Enable Expiration for Individual Secrets:
   - Once expiration is enabled for a template, it applies to all secrets created using that template.
   - Users with Owner permission can set custom expiration intervals for individual secrets via the Overview tab on the Secret View page.

Managing Expired Secrets

- **Forcing Expiration**: Users can manually force a secret to expire immediately by clicking `Expire Now` on the Secret View page.
- **Resetting an Expired Secret**: Change the field that has expired to reset the expiration interval.
- **AutoChanging an Expired Secret**: Enable AutoChange to allow Secret Server to automatically change the password when it expires. This requires enabling RPC and configuring it on the individual secret.
- **Retry Interval**: Secret Server checks for expired secrets every minute. If a password change attempt fails, it will retry based on the template's retry interval, which defaults to one hour.

Forcing Expirations

To force expiration:

1. Navigate to the **Secret View** page for the desired secret.
2. Click the **Expire Now** button. This forces the secret to expire immediately regardless of the interval setting. The expiration date displays "Expiration Forced."

Setting up Secret Templates for Secret Expiration

To set up expiration on a secret, you must first enable expiration on the template from which the secret is created.

To enable secret expiration for a secret template:

1. Navigate to **Admin > Secret Templates**.
2. On the **Secret Templates** page, click the template name. The page for that template appears.
3. Scroll down to the **Template Expiration** section.
4. Click the **Edit** button.
5. On the **Secret Template Designer** page, click on the **Change** link. The **Expiration Enabled?** check box appears.
6. Click to select the check box. Two more fields appear.
7. Type the number of days desired in the **Days until Expiration** text box.
8. Click the Change Required on dropdown list to select which parameter you want updated upon expiration. Your choices are:
   - Computer
   - Domain
   - My Server List
   - Password
   - Server
   - Username
9. Click the **Save** button.

You can override the interval setting for individual secrets.

Enabling expiration for a template enables expiration for all the secrets that were created using that template.

Resetting Expired Secrets

To reset an expired secret, you must change the text field that has expired and is required to change. For example, if the text field set to expire is the password text field and the current password is "asdf," then a change to "jklh" resets the expiration interval and thus removes the expiration text on the Secret View page.

If you do not know which text field is set to expire:

1. Go to the secret template that the secret was created from.
2. Navigate to **Admin > Secret Template**.
3. Select the template.
4. Click the **Edit** button.
5. On the next page, click the **Change** link. In the **Change Required On** text box you can see the text field that is set to expire.

Setting up Secrets

Once you enable expiration for the template, expiration is also enabled for secrets that were created using that template as well as secrets created in the future. The Expiration tab appears on the Secret View page and requires the user to have Owner permission on the secret.

To set a custom expiration at the secret level, you adjust the expiration interval for the secret by clicking the **Expiration** tab in the **Secret View** page. There, you can set the secret to expire using the template settings (default), a custom interval, or a specific date in the future.

Secret Navigation Slideout

The Secret Navigation Slideout is a set of useful links to secret. Its appears on the top navigation bar.

Click the tab and the slideout appears:

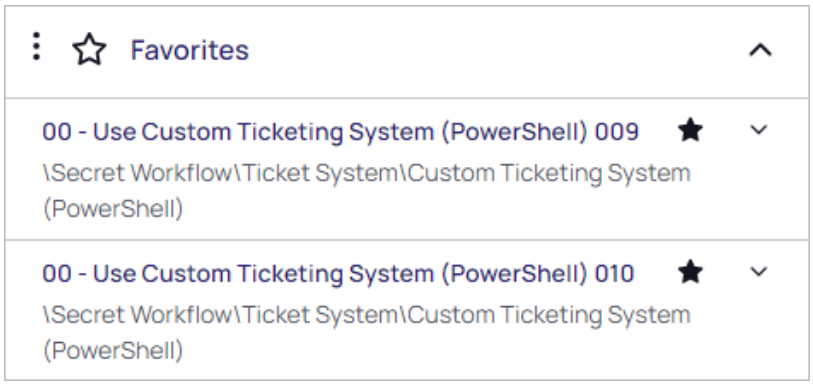
There are four dropdowns:

Favorites

These are the secrets you set as favorites by clicking the star icon on the secret's row on the All Secrets table.

Dropdown Quick List

When you click the dropdown, you see a list of your favorite secrets:



The initial view shows the folder path to the secret and a "favorite" toggle icon. You may need to hover over the blank space to see an unselected toggle.

> Some changes may require you to refresh the list by clicking the stacked dots icon.

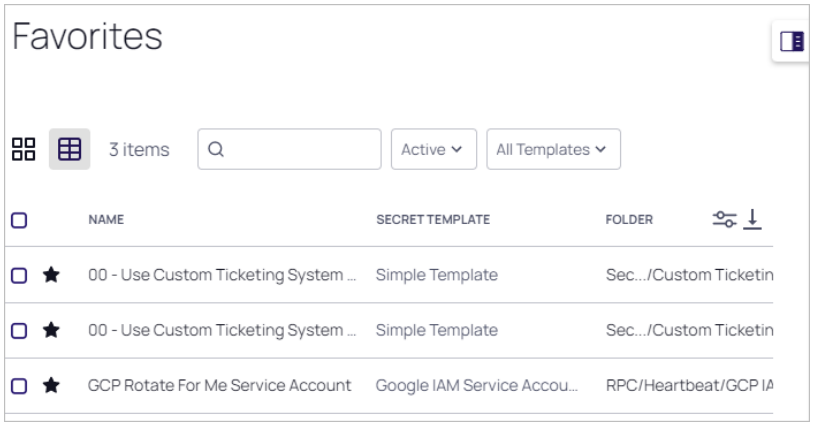
Each list item has its own dropdown. When you click it, a set of information about the secret appears, including a list of users or roles with access to (shared with) the secret.

> You may also have to enter a comment or check out the secret.

Most of the datums can be copied with a single click of the copy icon next to each. You can hover the mouse pointer over the "Shared with" list to see the type of access for that user or role, or you can click "See All" to see the entire list.

Favorites Page

Alternatively, in the slideout, you can click the Favorites title to navigate to the full Favorites page.



Recent

These are the secrets you have opened recently. The dropdown provided for each list item is exactly the same as that provided in "favorites," as is accessing the Recent page via clicking the title.

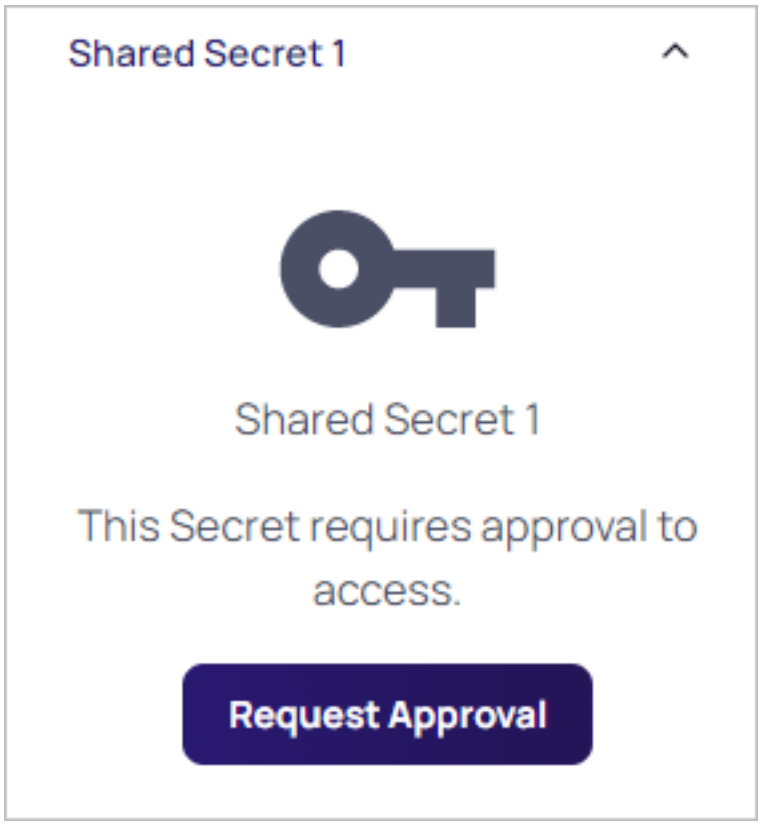
Most Used Secrets

These are the secrets you have opened the most often. The dropdown provided for each list item is nearly the same as that provided in "favorites," as is accessing the Recent page via clicking the title. The only difference is each list item states the time since last access.

Shared with Me

These are the secrets others have shared with you. No folder path is provided.

The dropdown provided for each list item provides a shortcut to request approval for the secret:



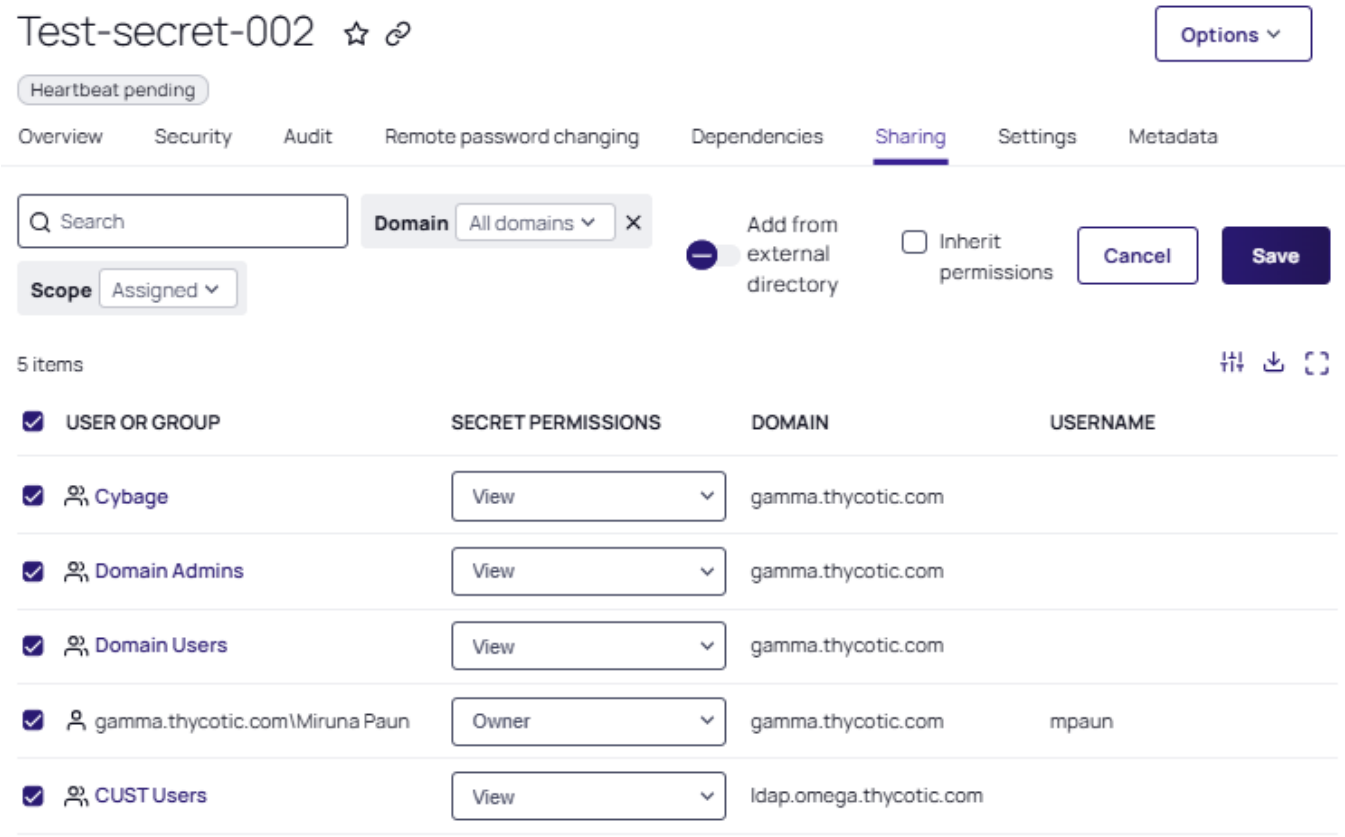
Secret Permissions

For information technology teams, securely sharing passwords is crucial. Due to the sensitive nature of sharing secure information, Secret Server ensures shared passwords are tracked and guarded via permissions.

> Depending on your configuration, folder settings can affect the permissions of secrets contained in that folder (and its subfolders). Secrets and folders are not visible to users who do not have at least the View Secret (can see details) or the View Lists (can see a list of secrets) permissions. See "Folder Permissions" on page 1059 for more information.

To simplify sharing, new secrets automatically inherit the settings from the folder they are stored in. The **Inherit Permissions** checkbox (found in the **Sharing** tab of the secret's page) is enabled by default. Secrets inherit all the parent folders' sharing settings. As long as this checkbox is selected, you cannot modify the permissions for the secret. For more on folder security, see the "Folders" on page 1059 section.

Secrets can be shared with groups or individual users. The **Sharing** tab in a secret's page allows for permissions and access to be configured:



There are four permission levels when sharing secrets with another user or group:

- **List**: The user may see a secret in a list, such as a list returned by running a search, but will not be able to view any more details about that secret or edit it.
- **View**: The user can see all data of the secret, such as username, password, metadata, permissions, auditing, history, and security settings.
- **Edit**: The user can edit the secret data as well as deactivate secrets. This permission also allows users to move the secret to another folder unless the **Inherit Permissions from Folder** setting is turned on, in which case the user needs the Owner permission to move the secret.
- **Owner**: The user may change all of the secret's metadata.

Password text-entry fields are not visible if a secret has a launcher and the **Hide Launcher Password** setting is on or the user does not have the **View Launcher Password** role permission.

The API calls use role permissions which are defined in the **tbRolePermission** table. In **tbFolderACL**, roles are not used, and it is less specific about the source, avoiding duplication.

Secret Templates Overview

Secret templates in Secret Server are pre-configured structures that define the fields, launchers, and remote password changers for different types of secrets. These templates simplify the management and automation of secret-related tasks. Here are some key aspects and examples of secret templates:

General Features

- Fields: Secret templates determine the specific fields that will be available for a secret, such as username, password, private key, and more.
- Launchers: Templates can include launchers that facilitate automated login or connection to remote systems.
- Remote Password Changers (RPC): Templates are pre-configured with password changers that can automatically update passwords on remote systems when a secret expires or on a defined schedule.
- Customization: Administrators can view, modify, and manage secret templates through the Secret Server administration panel.

Examples of Secret Templates

Oracle Account Secret Template:

- Purpose: Used for managing Oracle account secrets.
- Features: Includes fields specific to Oracle accounts and is configured with an Oracle-specific password changer.
- Usage: Automatically changes Oracle account passwords when a secret expires.

Windows Account Secret Template:

- Purpose: Used for managing Windows account secrets.
- Features: Includes fields specific to Windows accounts and is configured with a Windows-specific password changer.
- Usage: Automatically changes Windows account passwords when a secret expires.

SAP Account Secret Template:

- Purpose: Used for managing SAP account secrets.
- Features: Includes fields specific to SAP accounts and is configured with an SAP-specific password changer.
- Usage: Automatically changes SAP account passwords when a secret expires.

Azure Active Directory Secret Template:

- Purpose: Used for managing Azure AD account secrets.
- Features: Includes fields specific to Azure AD accounts and is configured with an Azure AD-specific password changer.
- Usage: Automatically changes Azure AD account passwords when a secret expires.

Unix Account (SSH) Secret Template:

- Purpose: Used for managing Unix account secrets with SSH.
- Features: Includes fields for private keys and passphrases, and is configured with a Unix-specific password changer.
- Usage: Automatically changes Unix account passwords and updates public keys when a secret expires.

Managing Secret Templates

- Activation and Deactivation: Secret templates can be activated or deactivated as needed.
- Mapping: Templates can be mapped to specific RPCs to ensure the correct password changer is used.
- Administration: Templates are managed through the Secret Server administration panel, where they can be created, edited, and assigned to secrets.

List of Built-in Secret Server Templates Secret Server

Secret Server includes many pre-configured secret templates:

Built-in Secret Templates Available Out-of-the-Box

For details about specific RPC secret templates, see "Included RPC Templates" on page 951.

- Active Directory Account
- Amazon IAM Console Password
- Amazon IAM Key
- Azure AD Account
- Bank Account
- Cisco Account (SSH)
- Cisco Account (Telnet)
- Cisco Enable Secret (SSH)
- Cisco Enable Secret (Telnet)
- Cisco VPN Connection
- Combination Lock
- Contact
- Credit Card
- DevOps Secrets Vault Client Credentials

- Generic Discovery Credentials
- Google IAM Service Account Key
- Healthcare
- HP iLO Account (SSH)
- IBM iSeries Mainframe
- MySql Account
- OpenLDAP Account
- Oracle Account
- Oracle Account (TCPS)
- Oracle Account (Template Ver 2)
- Oracle Account (Walletless)
- Password
- Pin
- Product License Key
- SAP Account
- SAP SNC Account
- Security Alarm Code
- Social Security Number
- SonicWall NSA Web Admin Account
- SonicWall NSA Web Local User Account
- SQL Server Account
- SSH Key
- Sybase Account
- Unix Account (Privileged Account SSH Key Rotation - No Password)
- Unix Account (Privileged Account SSH Key Rotation)
- Unix Account (SSH Key Rotation - No Password)
- Unix Account (SSH Key Rotation)
- Unix Account (SSH)
- Unix Account (Telnet)
- Unix Root Account (SSH)
- VMware ESX/ESXi
- WatchGuard
- Web Password

- Windows Account
- z/OS Mainframe

Managing Secret Templates

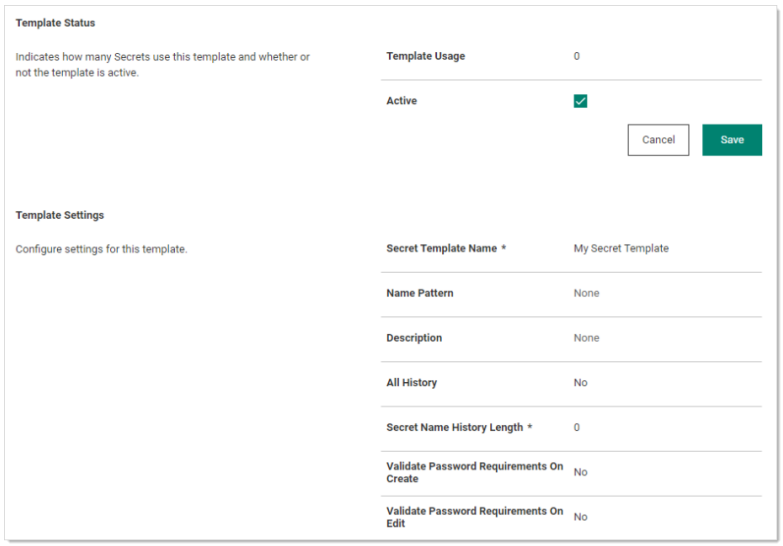
Including these topics:

- Changing a Secret's Template
- Configuring Secret Template Permissions
- Creating and Editing Custom Password-Exclusion ...
- Creating or Editing Secret Templates
- Field Slug Names
- Managing Specific Templates
- Secret Template Fields
- Secret Template List Fields
- Secret Template Settings
- Template Character Sets
- Template Naming Patterns
- Template Password Requirements

Activating and Deactivating Templates

If a template is no longer relevant or outdated, it can be inactivated. This can be done from the specific template's Secret Template Edit page.

1. Go to **Admin > Secret Templates**.
2. Click the template name in the **Secret Templates** column. That template's page appears on the General tab.
3. Click **Edit** next to Template Status. The Secret Template Detail Status section enters edit mode:



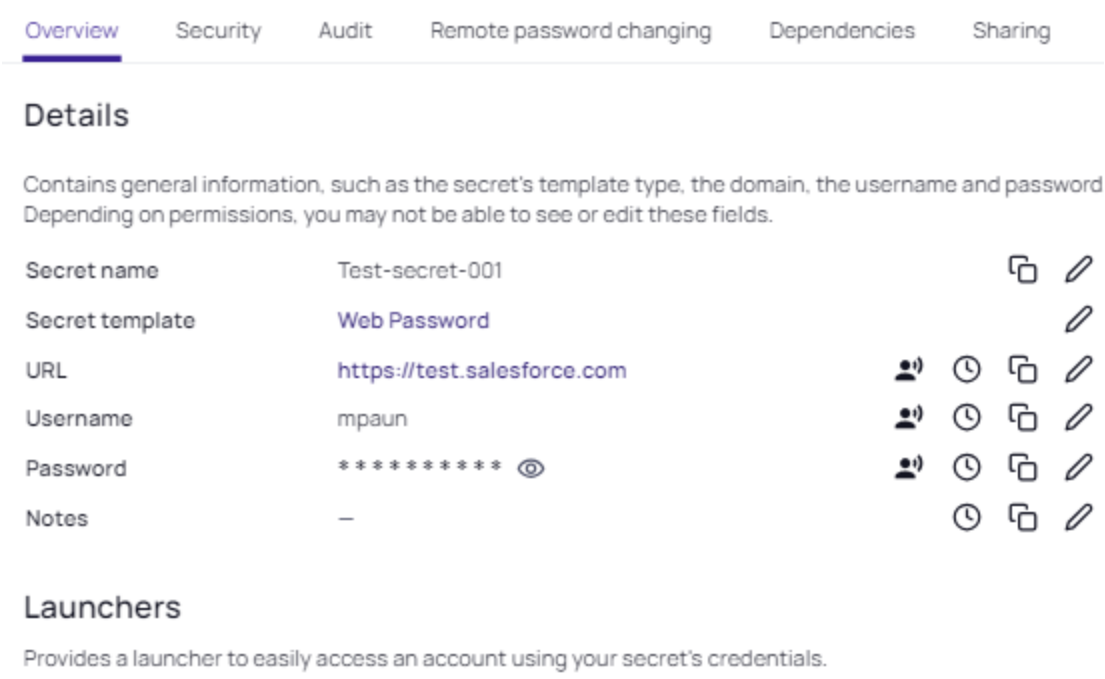
4. To deactivate a template, uncheck the **Active** check box.
5. To activate a template, check the **Active** check box.

Templates can also be inactivated in bulk from the Manage Secret Templates page. Click the **Active Templates** button to navigate to the Set Active Secret Templates page. This screen displays all the secret templates in Secret Server. Each secret template can be set as active or inactive. Once the secret templates are chosen as active or inactive, then saving changes brings the secret templates into effect immediately. Inactivating a secret template does not inactivate any secrets using that secret template–those secrets still exist, but users are not able to create new secrets using an inactivated secret template.

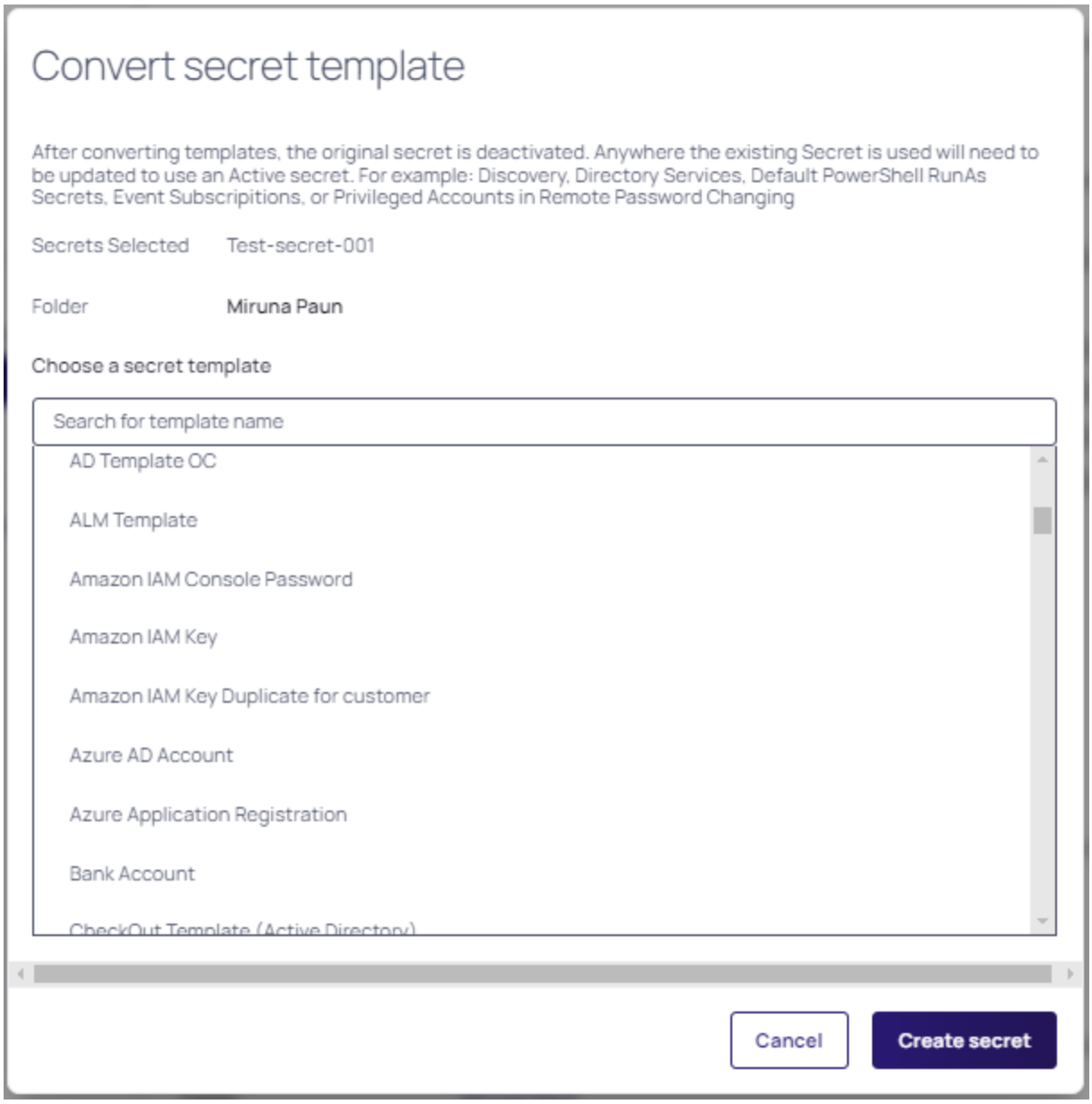
Changing a Secret's Template

To convert secrets from one secret template to another, do the following:

1. Search for and view a secret. Once in the **Overview** page, click on the **Convert Template** button. This button is located under the **Details** section in the form of pencil next to the Secret template field:



2. A pop up for **Convert secret template** appears. Search for the template you want to convert to from the list, and click **Create secret**:



3. The page with all the fields for the new template appears. Map each text-entry field your current template has to a new value. These values are: **New Secret Name**, **URL** (domain), **Username**, **Password**, **Notes**.
   a. If you want to remove the value for a text-entry field instead of converting it, select the `<Remove>` option in the list for that text-entry field.
   b. Click **Create secret**, when done, and then you can choose a new folder to move the secret to.

The Convert Template button is only available to users and groups with the "Owner" permission to the secret you with to convert. You can check what permission the secret has under the **Sharing** tab.

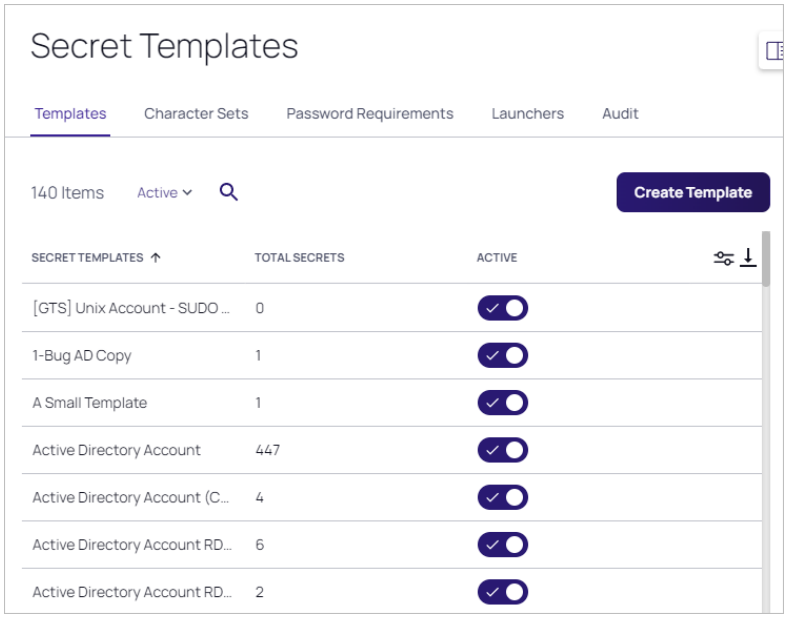
To preserve audit data, when a secret is converted from one type to another, the old secret is deleted, and a new secret is created. An admin can view the old secret by searching for deleted secrets on the dashboard.
A user needs "Add Secret", "Copy Secret", "Deactivate Secret", "Edit Secret", "Own Secret" and "View Secret" role permissions in order to convert a secret to a new template.

Configuring Secret Template Permissions

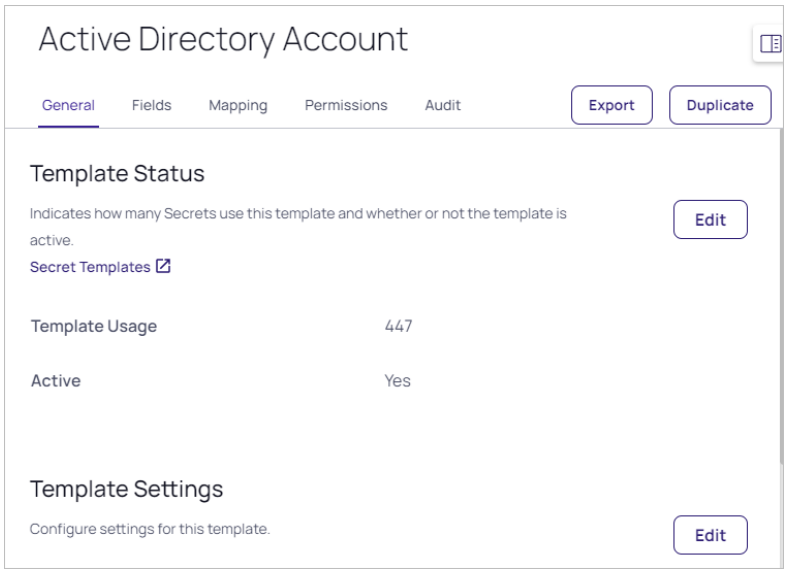
You can assign users and groups to specific secret templates so they can either manage or create secrets based on those templates. This allows you to have more granular control over what secret templates are seen by users and groups when they are managing the templates or creating secrets. To configure permissions:

Secret template access is dependent on configuration in **Groups** / **All Vault Users** / **Secrets**. Without modifying this, the group or user assignment on secret templates changes nothing.

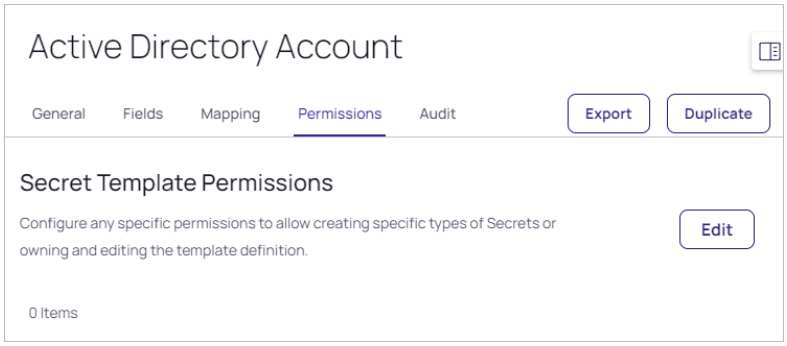
1. Select **Admin > Secret Templates**. The Secret Templates page appears:



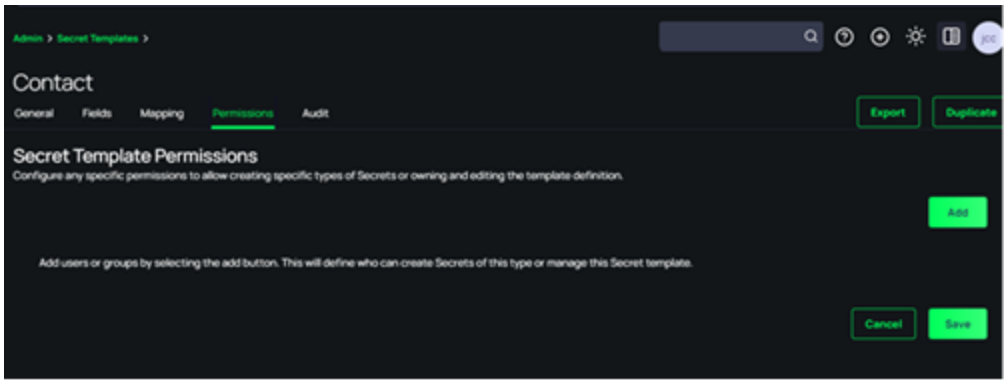
2. Click the name of the template for which you wish to configure permissions. The Secret Template Detail page appears on the General tab:



3. Click the Permissions tab:



4. Click the **Edit** button in the **Secret Template Permissions** section. The Secret Template Permissions section enters edit mode:

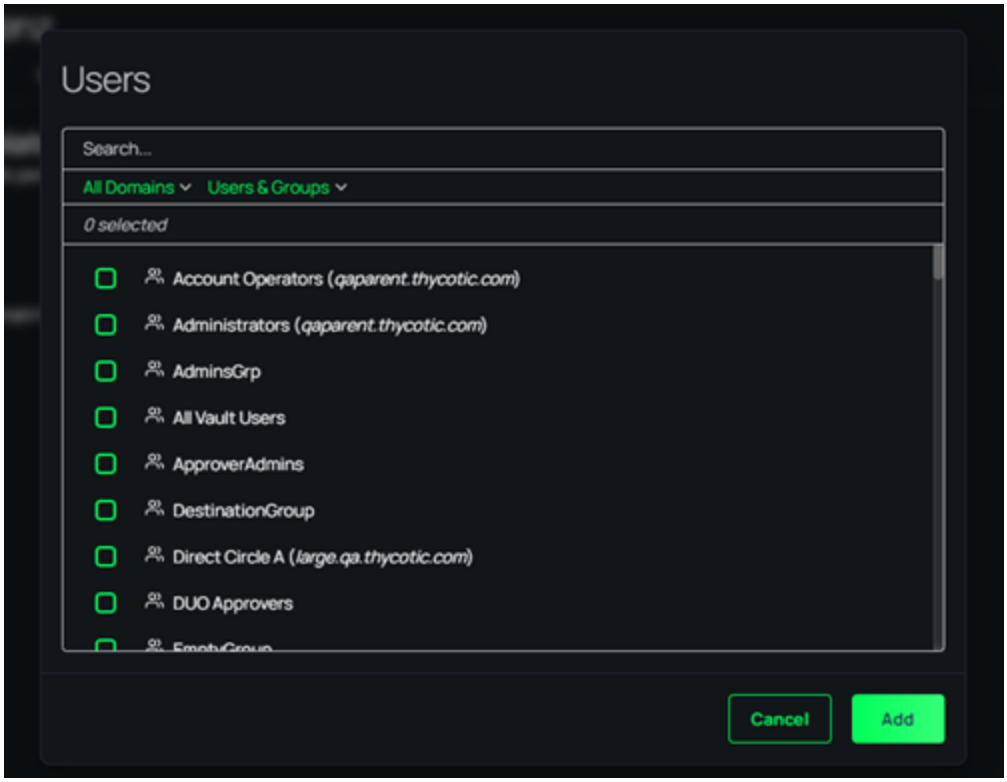


5. To change the permissions of a user or group that has already been assigned permissions, select a new permission from the **Administrator** drop-down list. Template Create Secret allows a user or group to create

secrets based on the selected secret template. Template Owner allows a user or group to edit a secret template and create secrets based on the selected secret template. By default, the All Vault Users group that targets all users of Secret Server can create secrets based on any secret template.

> User's secret template permissions are based on the permissions directly assigned to them, as well as the permissions assigned to all of the groups the user belongs to. If a user or group does not have Template Create secret or Template Owner permissions, they are unable to create a secret based on that secret template or see that it exists in Secret Server.

6. To add permissions for a new user or group, click the **Add** button. The Users popup appears:



7. Click to select the check box for the user or group.
8. Click the **Add** button.

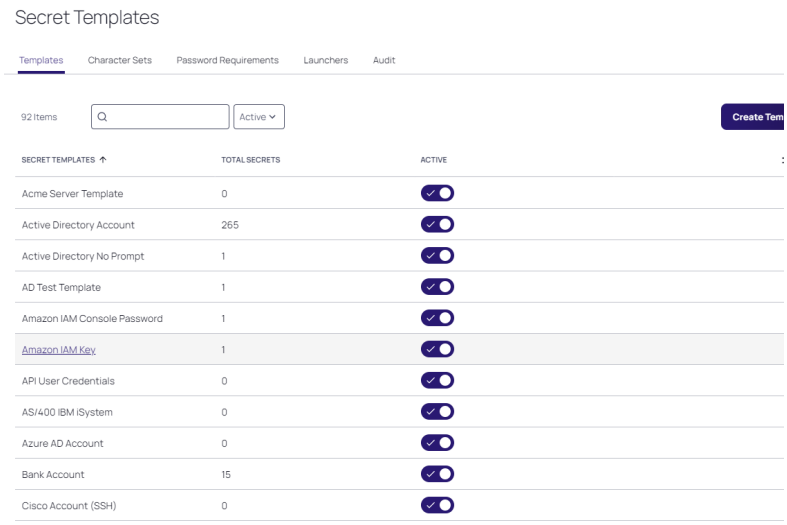
Creating and Editing Custom Password-Exclusion Dictionaries

A custom password-exclusion dictionary is a list of words that you do not want users to choose as part of a password, for example, your company name. The dictionary becomes an option when creating or editing a password requirement object. Those, in turn, appear as options when creating a secret template. Finally, when a secret is created based on that template, the words in the dictionary are not allowed when creating a password (the "weak" warning appears).

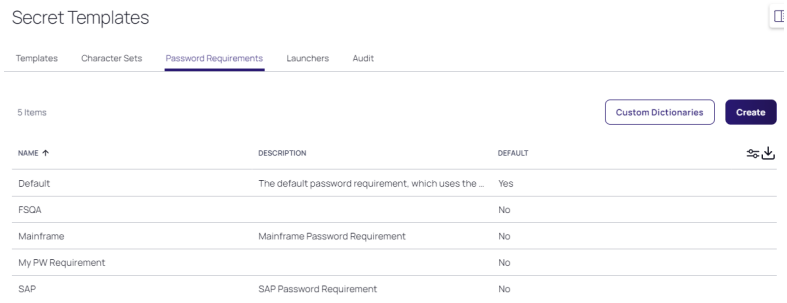
Creating a Custom Dictionary

To create a new custom password-exclusion dictionary for use by secret templates:

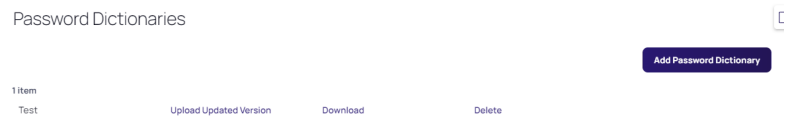
1. Create a text file containing the words you want to exclude, one word per line. These words cannot be used as part of a password on applicable secrets.
2. Go to **Admin > Secret Templates**. The Manage Secret Templates page appears:



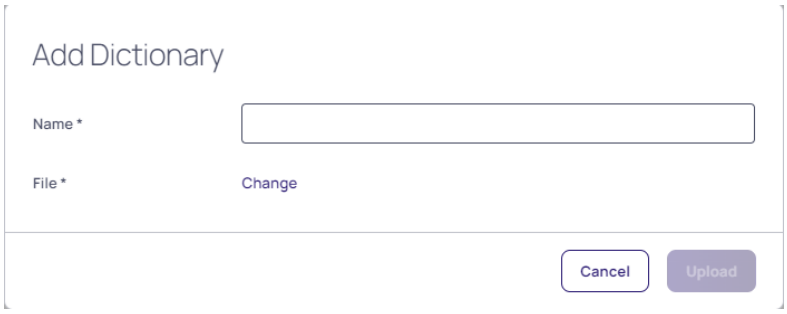
3. Click the **Password Requirements** button. The Password Requirements page appears:



4. Click the **Custom Dictionaries** button. The Password Dictionaries page appears:

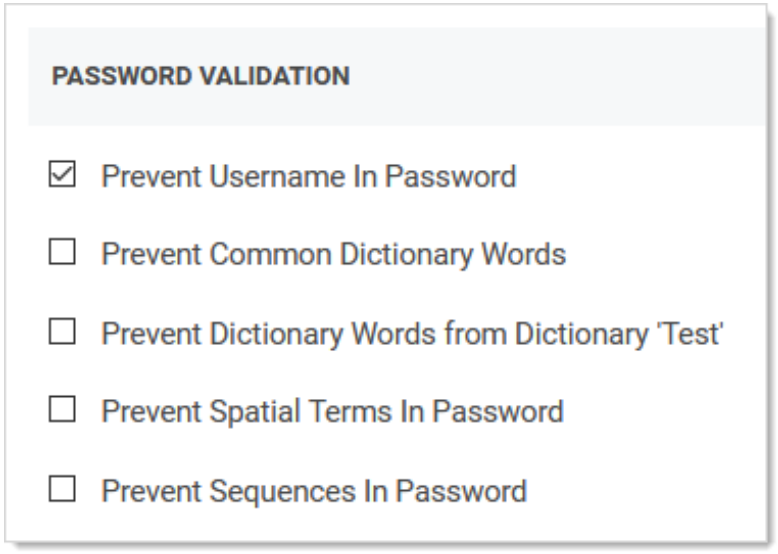


5. Click the **Add Password Dictionary** button. The Add Dictionary popup page appears:



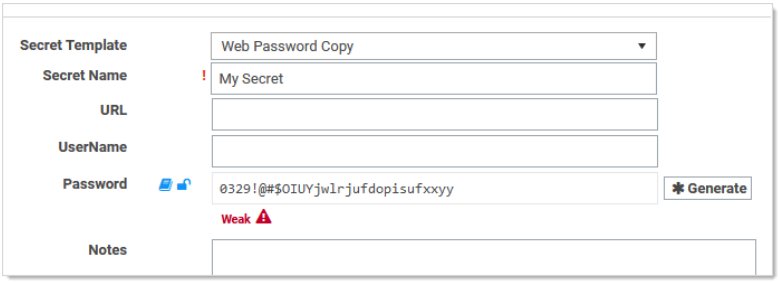
6. Type the name of the dictionary in the **Name** text box.
7. Click the **Change** link to locate your dictionary text file. The name of the file appears on the popup.
8. Click the **Upload** button. The popup disappears, and the file appears on the Password Dictionaries page:

Now, when defining a password requirement, the custom dictionary you created ("test") appears as a prevention option:

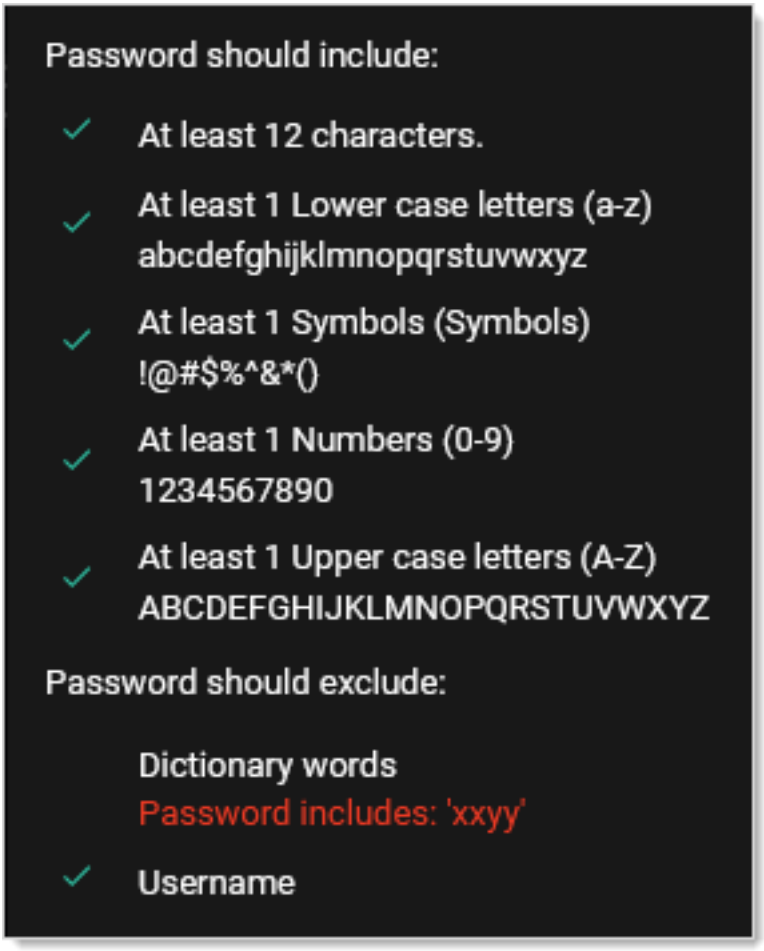


When a user attempts to include one of the excluded words in the dictionary in a secret based on the template using the password requirement, the "weak" warning appears and the user cannot save the password. For example, our dictionary contains the word (string) `xxyy`. The user enters a strong password that contains the string, and Secret Server rejects it anyway:

> The excluded words are not case sensitive. XXyY would have triggered a password rejection too.



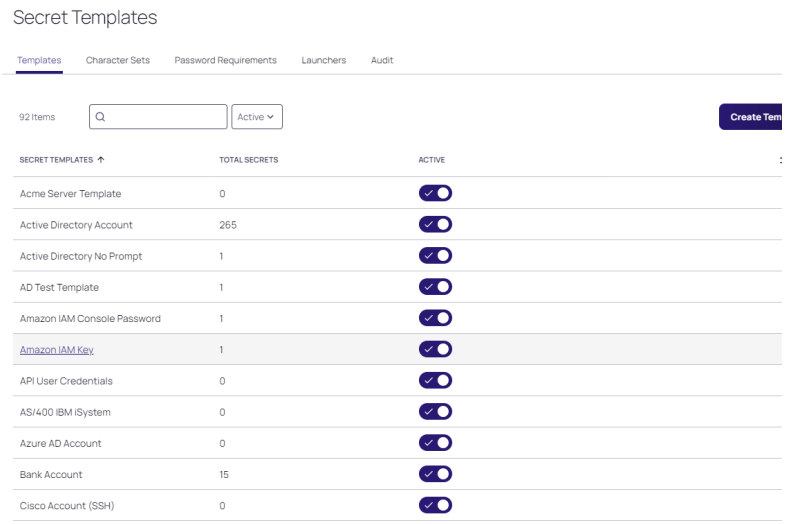
When you hover the mouse pointer over the password strength bar, the disallowed string appears in red:



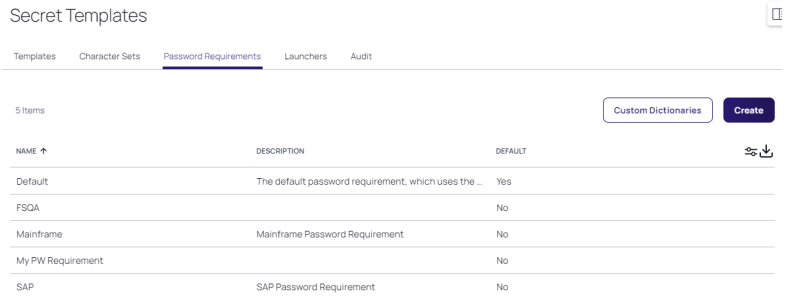
Editing a Custom Password-Exclusion Dictionary

To edit a custom password-exclusion dictionary for use by secret templates:

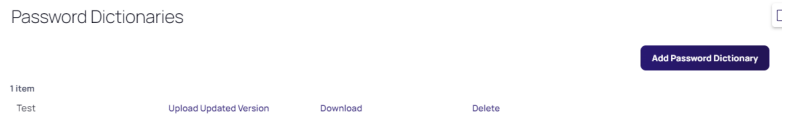
1. Go to **Admin > Secret Templates**. The Manage Secret Templates page appears:



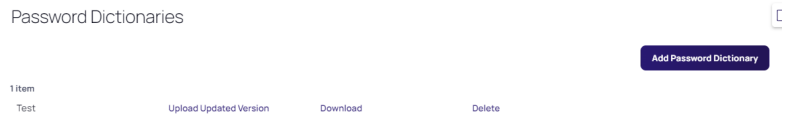
2. Click the **Password Requirements** button. The Password Requirements page appears:



3. Click the **Custom Dictionaries** button. The Password Dictionaries page appears:



4. Click the **Add Password Dictionary** button. The Add Dictionary popup page appears:



5. Click the **Download** link for the desired dictionary.

6. Save the file to your computer.
7. Edit the text file as desired. Do not change the name of the file.
8. Click the **Upload Update Version** link to locate and upload your dictionary text file. The existing dictionary, of the same name, is overwritten.

Creating or Editing Secret Templates

Editing Secret Templates

1. Select **Admin > Secret Templates** in the **Core Actions** section. The Secret Templates page appears.
2. Click the template name in the Secret Templates column. That template's page appears.
3. Click the desired tab for the configuration you want to change. See the Creating or Importing a New Template section for details.

Creating or Importing a New Template

Task 1: Creating the Template

1. Select **Admin > Secret Templates** in the **Core Actions** section. The Secret Templates page appears.
2. Click the **Create / Import Template** button. The Create Template pop-up page appears.
3. If importing the template, click to select the **Import XML** selection button.
4. Type the name of the new template in the **Template Name** text box.
5. Click the **Save** button. The new template's setup page on the General tab appears. The page provides all the options for configuring a secret template, as well as which text-entry fields appear on any secret created from that template.

Task 2: Adding General Settings and Setting an Expiration or One-Time Password

1. On the **General** tab, click the **Edit** link in the **Template Settings** section. The section becomes editable:
2. Edit the settings as desired. They include:
   - **Secret Template Name** text box.
   - **Name Pattern** text box. See "Template Naming Patterns" on page 1181.
   - **Description:** An optional description for the template.
   - **All History** check box: If this check box is enabled, Secret Server keeps all entries for viewing. This feature creates a record of every name used when a new secret is created.
   - **Secret name History Length** text box: If All History is disabled, Secret Server keeps this number of entries for viewing.
   - **Validate Password Requirements on Create?** check box: Ensure requirements are met on secret creation.
   - **Validate Password Requirements on Edit?** check box: Ensure requirements are met when editing secret.

- **Required Permission To Edit Password Change Configuration** dropdown list: Specify which permission is required on the password change configuration on a secret from this template.

3. Click the **Save** button.
4. Click the **Edit** link for the **Template Expiration** section. Secret expiration applies to one field of a secret template (most commonly the password field) and may trigger a password change for that secret if auto-change is enabled for RPC.
5. Click to select the **Expiration Enabled?** check box. Two additional controls appear.
6. Type the days till expiration in the **Days until Expiration** text box.
7. Click the Change Required On dropdown list to select the field to choose the field the expiration is applied to.
8. Click the **Save** button.Secret Server begins providing alerts if the secret text-entry field is not changed within the specified expiration requirements.
9. Click the **Edit** link for the **One Time Password** section if you want the secret to have a one-time password that the user must change.
10. Click to select the **One Time Password Enabled** check box. Additional controls appear:
11. Type or select the options.
12. Click the **Save** button.

Task 3: Defining Fields for the Template

Click the **Fields** tab to add template fields as desired. See "Secret Template Fields" on page 1176.

> To use a custom SSH RPC port, add a field named "Port" to your secret template. Empty port fields are equivalent to the default port, 22.

Task 4: Mapping Launchers and RPC Type

1. Click the **Mapping** tab to configure launchers and RPC.
2. Click the **Edit** button in the **Password Changing** section to enable RPC on secrets based on this template. This enables heartbeat, RPC, and configures the password changer type and fields. For details, see "RPC Overview" on page 904.
3. Click the **Add Mapping** button to add a secret launcher or extended mapping. The Add Mapping popup appears.
4. Click the **Mapping Type** combination list to search for or select a mapping type:

   Launchers:

   - Command Prompt
   - Custom PuTTY Launcher (Port Field on Secret)
   - Custom Launcher with Host Prompt
   - IBM iSeries Launcher
   - Mac Process - Default Client - No Prompt

- PowerShell ISE
- PowerShell Launcher
- PuTTY
- PuTTY With Port Prompt
- Remote Desktop
- SAP Custom Launcher
- SQL Server Launcher
- SQLPlus
- Sybase iSQL Launcher
- TextEdit (OSX)
- Website Login
- Windows Notepad
- z/OS Launcher

Extended Launchers:

> Extended Mappings allows you to tie a text-entry field value to a SS defined system type for additional functionality. For example, you may have a generic password secret template that has a username and password text-entry field. For purposes of looking up credentials, such as a ticket system authentication secret, Secret Server needs to know that actual type of the text-entry fields since the text-entry field name can be custom.

- OATH Secret Key: For password changing on the Amazon Root Account using the Web Password Changer. If you enter the OATH secret for two factor, SS generates the one-time password (OTP) automatically for password changing and heartbeat, allowing you to automate that while enforcing two-factor authentication on the AWS root credential.
- Regex List
- Remote Server SSH Key for Validation: Ensures the machine SHA1 digest for validating the machine connected to is correct.
- SSH Private Key: Defines which text-entry fields make up the SSH Key components of Private Key, Private Key Passphrase, and Public Key.
- Username and Password: Defines which text-entry fields contain the username and password.

The popup changes to accommodate your choice.

> A secret launcher launches applications on other machines and automatically logs on using credentials stored in Secret Server. In general, there are three types of launchers: RDP, SSH, and Custom. In addition to user convenience, launchers can circumvent users needing to know their passwords—a user can still gain access to a needed machine but it is not required to view or copy the password out of Secret Server. A Web launcher automatically logs into websites using the client's browser.

5. Click to select or type to search the desired dropdown lists.
6. Click the **Save** button.

Task 5: Adding Permissions

1. Click the **Permissions** tab. This defines who can create secrets of this type or manage this secret template.
2. Click the **Edit** button's dropdown and select **Add**. The Users list appears.
3. Type the name of the user or group you want to add in the **Search** text box. Note that the groups are by domain.
4. Click to select the user or group's check box for those you desire.
5. Click the **Add** button. The selected users or groups appear on the Permissions tab.
6. Click the dropdown list next to each to define if the user or group has the Template Create Secret or Template Owner permission. More than one owner is allowed.
7. Click the **Save** button the users or groups now appear in a small table, along with their roles (permissions).
8. To remove a user or group:
   a. Click the **Edit** link for the **Secret Template Permissions** section. The table of users and groups disappears, and the dropdown lists reappear.
   b. Click the dropdown list for the user or group you want to delete and select **<None>**.
   c. Click the **Save** button

Task 6: Viewing the Template's Audit Trail

1. Click the **Audit** tab to view activity on the secret template.

> You cannot drill down on the entries, but you can define what columns to see by clicking the slider icon on the right. You can also click the download icon to download a text file version of the table.

Settings for Specific Template Types

Oracle Account as SYS

Settings for an Oracle Account secret template to work with Oracle connecting as SYS in SysDBA:

- Set **Oracle Account** as the type.
- Set **Oracle Account (AS SYS)** as the password type.
- Create a secret based on the new template to test the template.

SQL Windows Authentication Account Secret Template and Launcher

Settings for an Active Directory template that is specifically for SQL:

> You can copy the existing AD template that you have. However, if you copy an existing template that has launchers attached to it, you may need to delete those launchers on the newly created template.

- Set **Active Directory** as the type.
- If necessary, create a field called **Server**.
- Add the following parameters for Windows settings (see "General Settings" on page 669):
  - Name: SQL Server Launcher - Windows Authentication
  - Active: Yes
  - Process Arguments: -E -S $Server ($Server should match the field name you created or observed earlier)
  - Run Process as Secret Credentials: Yes
  - Load User Profile: Yes
  - Use Operating System Shell: No
  - Use Additional Prompt (in General Settings): No

Secret Template Settings

These include:

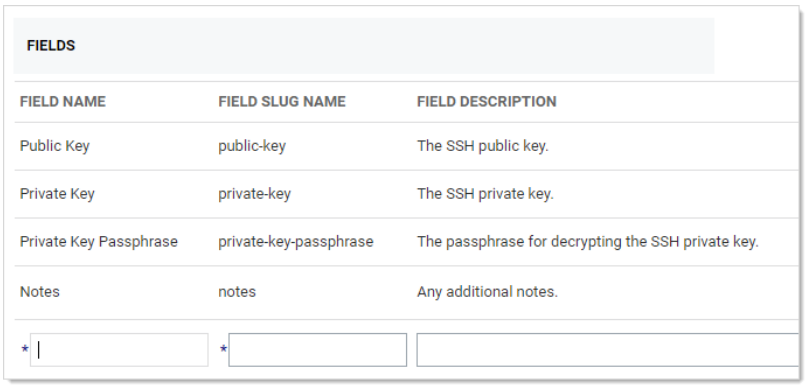
- "Field Slug Names" below
- "Secret Template Fields" on the next page
- "Introduction" on page 1178

Field Slug Names

A *field slug name* in Secret Server is a unique human-readable identifier for a data field in a Secret Server template. The field slug name is available for integrating with third-party applications via API calls. Slug names are programmatically available for API calls but are not visible to template users (secret creators). Instead, they are displayed as references in secret templates.

> If you are not planning to access Secret Server with an API, slug field names are not for you—leave the suggested name as is.

Figure: Field Slug Name in a Secret Template



FIELDS

FIELD NAME	FIELD SLUG NAME	FIELD DESCRIPTION
Public Key	public-key	The SSH public key.
Private Key	private-key	The SSH private key.
Private Key Passphrase	private-key-passphrase	The passphrase for decrypting the SSH private key.
Notes	notes	Any additional notes.
*	*	

Field slug names are automatically generated, based on the field name, when the field is created. For example, "User Name" became "user-name." Characters that are potentially problematic for programming, such as spaces, are swapped out. The automatically generated name is unchangeable by human users, unlike the field name. If API calls were based on the field name, human users with access to the template could break those calls, simply by changing the name.

With Secret Server 10.7.X+, The generated field slug names are now user-definable. You can edit the generated names to:

- Conform to a naming convention used in your API calls.
- Maintain the same name for a field across secret templates to simplify coding by developers.

The only requirement is that each slug field name is unique to that template.

If you are wondering how Secret Server internally uniquely identifies fields, there is an internal ID that is not accessible by users or APIs. It is not available read-only (for API use) because we want to futureproof integrations from internal changes to Secret Server.

The user-definable field slug names are also automatically generated when you upgrade from a version of Secret Server that did not have user-defined field slug names. If there are two fields with the same field name, the second (and later) generated field slug name has an incremented number appended to it.

Secret Template Fields

If you want to programmatically manipulate fields, see "Field Slug Names" on the previous page.

To use a custom SSH RPC port, add a field named "Port" to your secret template. Empty port fields are equivalent to the default port, 22.

Field Types

Template fields can be specified as one of several different types to enhance customization:

- **File:** File attachment link. File attachments are stored in the Microsoft SQL Server database.
- **General List:** Preconfigured selectable list for launcher enhancement or general use. See "Introduction" on the next page.
- **Notes:** Multi-line text-entry field.
- **Password:** Password type text-entry field.
- **Text:** Single-line text-entry field.
- **URL:** Clickable hyperlink.
- **URL List:** Preconfigured selectable list for general use. See "Introduction" on the next page.

Editing Fields

The secret template designer provides several settings to customize secret template text-entry fields:

- To add a secret text-entry field, fill out the values and click the **+** button.
- To delete a text-entry field:
  1. Click **Add Field.**
  2. Click on a field name to edit the details of that field.
  3. From the template details you can edit the active status.
  4. Only active fields will appear in the template so to remove a field from showing flag it as not active.
- To edit a text-entry field, click the  icon. Click either the  icon to save or the **X** icon to discard the changes.

Text-Entry Field and Control Settings

The settings available for text-entry fields are:

- **Field Name:** Name of the text-entry field. This name is used for the Create New drop-down list on either the Dashboard's Create Secret Widget or Home page.
- **Field Slug Name:** A unique identifier used in API calls and other interactions. The slug allows the display name to change without breaking interfaces to fields.
- **Description:** Description of the text-entry field.
- **Type:** Type of the text-entry field. See below for a description of the different text-entry fields.
- **Is Required:** Whether the text-entry field should require a value. These check boxes are checked for correct content when the user attempts to create this secret. A validation error is displayed if not entered correctly.
- **All history:** Check to save all history of this field.
- **Searchable:** Whether that text-entry field should be indexed for searching. By default, passwords are not indexed. File attachments and history cannot be indexed for searching.
- **Edit Requires:** Minimum permissions on the secret needed in order to edit the value on the secret. The options are Edit, Owner and Not Editable. This enables the secret text-entry field to be locked down at a more granular level than other text-entry fields on the template.

- **Viewing requires edit:** If checked, this text-entry field is not displayed to users when viewing the secret. The text-entry field is only be displayed when the secret is in Edit mode.
- **Expose for Display:** If checked, this text-entry field is available to be displayed as a Custom Column on the Secret Server Dashboard.
- **Dropdown options:** Add the values to appear as dropdown for the text fields.

> All text-entry fields that are set to "Expose for Display" are **not** encrypted in the database. Only check this value if the secret text-entry field data is not considered privileged information.

The order of the text-entry fields in the Template Designer grid is the same as those that appear when the user views or edits a secret created from the template. The order can be modified through the up and down arrows on the grid.

Default values can be specified on each text-entry field by clicking the edit defaults  button . These added values appear as a list on any secret created from this template.

Secret Template List Fields

Overview

Introduction

With secret template list fields, administrators can create new lists that can be shared by multiple secrets. Clicking on an existing list goes to the details page for that list where the user can set the list's name, description, and the options available in the list.

You can optionally group list options by category, which make using very large lists easier. For instance, a list of machines might have the machines categorized by function, such as "Web Server" or "Database Server." You could also use categories for locations, such as "London," "New York," or "Tokyo."

List categories are displayed on the secret and on the launcher dialog with the options sorted alphabetically within categories, which are also sorted alphabetically. Options can be duplicated in multiple categories and will show up in each one. In addition to manually adding categories and options, you can upload a file containing the list options. Teams (Admin > Teams) – The team details page

In general, there are two types of list fields:

- Allow Lists: Display a searchable drop-down of the server names or IPs entered in the list for the user to select from. Only allows entries from the list to be used.
- Deny Lists: Allow the user to enter any server name or IP, but checks against the deny list and prevents connecting to entries on that list.

> If both are set, the user sees a dropdown of all items in the allowlist that are not also in the deny list. This setup can be useful when using the same allow list on multiple secrets but where you might not want some of those servers to be used on some secrets. Using both fields means that the customer does not need to create separate versions of almost identical allow lists and can instead just choose to restrict some.

Comma-Delimited Lists

There are two types of list filtering in Secret Server: The above mentioned list filtering and assigning a text or notes field on the secret as a comma-delimited list of server names. The former has the benefit of being shared between secrets, but the latter is useful as a one-off on a single secret. A comma-delimited list can be either an allow or a deny list.

Adding a New List Field

Task 1: Create the List

1. Go to **Admin > Categorized Lists**. The Lists page appears.
2. Click the **Create List** button. The Create List popup appears.
3. Type the name in the **Name** text box.
4. (Optional) Type a description in the **Description** text box.
5. Click the **Save** button. The configuration page for the new list appears:
6. Click the **List Options**tab.
7. Click on the expand dropdown button next to the **Create option** button and select **Create Category** from the dropdown. The Create Category popup appears.

   > If you want a list with no categories, choose **Uncategorized** for category, and follow these same instruction for adding options.

   > You can also create categories from a comma-delimited list in a text file. Select the **Add** button and select **Add from File**. This can be either a list of options, one option per line, or a list of comma-delimited values in the format option,category with one pair per line. Files can also combine these formats, and any line without a comma will be treated as a option without a category.Type the name for the category in the **Category** text box. We typed "Manhattan."

8. Enter the Category name and click **Save**. The category name now appears in the category dropdown list.
9. Add another category the same way.
10. Click **Create Option**. The Create Option popup appears.
11. Type the name for the Option in the **Option Name** text box.
12. Click to select the category the new option will belong to in the **Parent Category** dropdown list.
13. Click **Save**. The new option appears in the list.
14. Add another option the same way.
15. Repeat the process for the other category you created.
16. For future reference, an alternative method exists–click one of the options. A sidebar appears:

- Update Option: Rename the option
- Move to Category: Move the option to another category in the same list
- Delete Option: Remove the option from the category.

For now, we will not use any of them.

17. You now have a new categorized list available for secrets (via a secret template with the list).

If you ever want to view past changes to a list or category, click the Audit tab for the list.

Task 2: Create a Template Using the List

1. Go to **Admin > Secret Templates**. The Secret Templates page appears.
2. Click **Create/Import Template**. The Create Template pop up appears.
3. Type the template name in the **Template Name** text box.
4. Leave the option button set to **New**.
5. Click **Save**. The Secret Template Designer page for that new template appears.
6. Click the **Fields** tab.
7. Complete the following steps for each field:
   a. Click **Add Field**. The Add Field popup appears.
   b. Type a name in the **Name** text box for the first (and currently only) field.
   c. Click the **Type** dropdown list for the field and select **List**.
   d. Click **Save** button. The new field appears in the table.
8. Click the name of a new field in the list. Its configuration page appears.
9. (Optional) Click the **Edit** button in the **Template Field Details** section to further customize the field.
10. Click the **Secrets** button in the main menu to return to the Secret Server dashboard.

Task 3: Create a Secret Based on the Template

1. Click the **Create secret** button. The Create new secret popup appears.
2. In the **Choose a secret template** list, select the secret template you just created. Create New Secret popup updates to reflect your chosen template.
3. Note that one of the dropdown lists has the same name as the list field you created earlier. Click it, and you see the list categories you created. The list is now available for that secret's launcher.
4. When a secret gets created with a list template, a tab on the secret gets created called **List Fields**. You then need to navigate to the **Mapping** tab and add the created list to the launcher restrictions by editing the **Fields** field.

Template Character Sets

Character sets are a collection of distinct characters that are used in password requirements and password rules. Custom sets can be created, and both ASCII and Unicode are supported. For more information on setting up compliance checks and password generation standards, see "Template Password Requirements" below. The five standard character sets are:

- Lower Case (a-z)
- Upper Case (A-Z)
- Numeric (0-9)
- Non-Alphanumeric (!@#$%^&*())
- Default – Includes all the above

To manage character sets, click the **Character Sets** button on the **Administration > Secret Templates** page. Only character sets which are not currently used by a password requirement can be deleted.

Template Naming Patterns

Secret Server supports naming patterns for secret templates. Naming patterns are a way for administrators to maintain consistency for secret names and can help ease both browsing and grouping secrets by name. Patterns are created using regular expressions. Regular expressions are a formal set of symbols commonly used to match text to patterns. For example, the regular expression `^\w+\\\w+$`, allows `NTDOMAIN01\USER3454` but not `USER3454` on `NTDOMAIN01`.

> Regular expressions are beyond the scope of this document. They are very powerful and can get quite complex—books have been written on the topic. Microsoft offers a good overview at their Regular Expression Language Quick Reference Web page.

Template Password Requirements

Overview

A password requirement is a stored Secret Server object that defines the requirements on a password text-entry field to validate user-entered passwords or make auto-generated passwords conform to set specifications. You can have multiple password requirements, but only one can be set to the default.

A password requirement is made up of a minimum and maximum length, a set of characters, and optional rules such as "At least three upper-case characters" or "The first character must be lower-case". The default password requirement is 12 characters from the default character set with at least one upper-case, lower-case, numeric, and symbol character.

Creating a Custom Password Requirement

To create a new password requirement:

1. Click the **Settings** drawer in the main menu. The All Settings page appears.
2. Click the **Secret Templates** link in the **Secrets** Section. The Templates tab of the Secret Template page appears.
3. Click the **Password Requirements** tab.
4. Click the **Create** button. A popup appears.
5. Type the name of the new password requirement in the **Name** text box.
6. (Optional) Type a description of the new password requirement in the **Description** text box.
7. Click the **Minimum Password Length** spinner to select or type a minimum allowed password length.
8. Click the Character Set dropdown list to select a character set for the password. The out-of-the-box default is `abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()`.
9. If you want the password requirement to become the new default, click to select the **Is Default** check box.
10. Click to select the desired password no-no check boxes. The options are:
    - **Prevent Username**: Do not allow the username to be part of the password.
    - **Prevent Spatial Pattern**: Do not allow strings of characters based their order on the keyboard, such as `qwerty` or `asdfg`.
    - **Prevent Sequential Pattern**: Do not allow strings of characters based on their order in the character set, such as `abcd` or `5678`. Note: As character sets are case sensitive, `AbCd` is allowed.
    - **Prevent Dictionary Words**: Do not allow everyday English words in the password.
11. Click the **Save** button. The popup closes and the page for the new requirement appears (containing the choices you just made for the details and generation sections):
12. Scroll down to the **Password Validation** section.
13. Click the **Edit** button. The section expands.
14. Most of the validation rules are ones you have already set with these two exceptions, which you can now set:
    - **Prevent Dictionary Words**: Do not allow everyday English words in the password.
    - **Prevent Words from Dictionary ...**: Do not allow words that appear in the named dictionary. In our example, the dictionary is named "Test."
15. Go to the **Starting and Ending Character Validation** section.
16. Click the **Edit** button.
17. To require specific starting characters, click to select the **Require Specific Starting Characters** check box. Two hidden controls appear. This allows you to make rules such as "password must start with three symbols and end with two lowercase letters."

    
    "Start and end with" rules can decrease the password entropy (resistance to brute force attacks).

18. Type or click the spinner to set the number of required starting characters.
19. Click the **characters from** dropdown list to select the character set to draw the characters from.

20. Repeat the procedure for any desired ending characters.
21. Click the **Save** button. An edit button now appears for the Character Count Validation section.
22. To set character count validation rules:
    a. Click the **Edit** button for the **Character Count Validation** section. The section expands.
    b. Click the **Add Rule** button and select one of the following types:
        - **Minimum Required Characters Rule**: For the first rule type, type the number of characters and select what character set they must come from, for example, "Minimum 5 characters from Upper Case (A-Z)."
        - **Maximum Consecutive Characters Rule**: For the second rule, type the number of characters and select what character set they must come from, for example, "Maximum 5 characters from Lower Case (a-z)."
        - **Repeating Characters Rule**: Sets a limit on how many times any single character can appear in a password. You can set it anywhere between one and the maximum length of the password requirement. For example, the rule "At most 1 of the same character" means that any character can only appear one time in a password: `Bztyopz` is invalid because there are two `z` characters, and `Bztyopx` is valid because no character appears more than once
        - **Repeating Consecutive Characters Rule**: Sets a limit on how many times any single character can appear in a sequence in a password. You can set it anywhere between one and the maximum length of the password requirement. For example, "At most consecutively 2 of the same character" means any character can only appear one time in a password: `Bzty1fxeee` is invalid because there are three `e` characters at the end of the password. `Bzty11fxe` is valid because no character appears more than twice. Finally, `Bzetey1efxe` is valid, even though there are three `e` characters, because they do not appear next to each other.
    c. Once you create more than one rule, the **Minimum Required Character Count Rules** dropdown list appears. This allows you to set whether you want a minimum number of rules enforced from those you created or all of them.
    d. Create as many additional character count validation rules as you desire by clicking the **Add Rule** button and repeating the procedure.
    e. Click the **Save** button.
23. Review the **Password Rule Strength** section to see how strong your choices are and any recommendations for improvement. The two tests are:
    - **Entropy Score**: The difficulty of cracking the password in a brute-force attack.
    - **Total Strength Score**: An overall weighted measure of password strength for passwords generated by the password requirements. Any rule conflicts will appear in the recommendations section.

The explicit character rules cannot conflict with the implicit ones you created earlier or you will get an error when saving. For that reason, we suggest leaving the password requirements character set to the default. Carefully consider any other conflicts if you get an error.

To set a custom password requirement for a specific secret, use the "Customize Password Requirement" in the Security tab of a secret.

> You can enable or disable the validation of manually entered passwords at the secret template level via the "Validate Password Requirements on Create" and "Validate Password Requirements on Edit" settings.

> The "What Secrets Do Not Meet Password Requirements" report shows secrets containing a password that does not meet the password requirements set for its secret template.

> Password requirements cannot include rules with overlapping character sets. For example, if an attempt is made to add both a "Minimum of 1 upper-case" rule and a "Minimum of 3 Default" rule to a new password requirement, an error displays.

Assigning Requirements to a Secret Template

To assign requirements to a secret template:

1. Navigate to **Secret Templates.**
2. Select the template you wish to edit.
3. Navigate to the **Fields** tab.
4. Select the **Password** field.
5. Select Edit for the **Template Field Details** section.
6. For **Password Requirement**, select your desired requirement.

Managing Specific Templates

This section addresses specific secret templates, rather than templates in general.

Configuring SAP SNC Account Secret Templates

Introduction

The "SAP SNC Account" secret template is an expansion on the original "SAP Account" secret template. It takes advantage of SAP's Secure Network Communication (SNC), which is a protocol that encrypts communication between Secret Server and an SAP Server. The SAP SNC Account template includes all the original fields from the SAP Account secret, adding a few more as well.

New Template Fields

The following is an introduction to the new template fields (in addition to those also found in the SAP Account secret template):

> Please see the SAP .NET Connector 3.0 Programming Guide for additional information.

- **SNC Partner Name:** Matches the snc/identity/as value set in your SAP Server configuration.
- **SNC My Name:** For most SAP configurations, you can ignore this. See the connector programming guide for cases where it may be required.
- **SNC Quality of Service:** Dropdown list to select the service quality or protection used for SNC communication. Choose one of the following protection options:
  - Authentication Integrity (includes authentication)
  - Authentication Integrity Privacy (includes integrity protection and authentication)
  - Authentication Only
  - Default Protection
  - Maximum Protection
- **SNC Single Sign On:** Dropdown list to set to true if you wish to use single sign on. If you set this to false, you authenticate with your username and password on the secret.
- **X.509 Certificate:** Click the **Change** link to upload an X.509 certificate for authentication.

Server-Side Setup

Prerequisites

SAP Server Setup

Follow the latest SAP documentation for configuring the SAP server and your SAP users to use SNC. For example:

1. SSH into your SAP server.
2. Edit the configuration file: `/sapmnt/<SystemID>//profile/profilename.pfl`
3. Add the SNC settings to the end of this file. For example:

```
snc/enable = 1
snc/gssapi_lib = /usr/sap/NPL/SYS/exe/run/libsapcrypto.so
snc/identity/as = p:CN=vhcalnplci,OU=Test,O=Thycotic,C=US
snc/accept_insecure_cpic = 1
snc/accept_insecure_gui = 1
snc/accept_insecure_r3int_rfc = 1
snc/accept_insecure_rfc = 1
snc/permit_insecure_start = 1
snc/extid_login_diag = 1
snc/extid_login_rfc = 1
snc/data_protection/min = 1
```

4. Verify that the library file path exists on your server (make sure that `libsapcrypto.so` is actually in that directory).
5. Reboot your server.
6. When the server is finished, reconnect and restart the SAP server with these commands:

```
su npladm
startsap all
```

SAP NCO Files

As with the original SAP Account template, you include the `SAPNCO.dll` and `SAPNCO_UTILS.dll` files in your Secret Server or distributed engine installation. See "SAP Heartbeat and Password Changing" on page 1026 for more information.

SAP Cryptographic Library

In addition to the SAP NCO DLL files, you need to obtain the SAP Cryptographic Library. This should include the library DLL (`sapcrypto.dll`), the license ticket, and the configuration tool (`sapgenpse.exe`). Add the DLL file to your Secret Server or distributed engine installation following the same steps as the SAP NCO files. For more information on this library, see the SAP Identity Management Configuration Guide .

SAP Server Certificate

1. Open SAP Trust Manager (STRUST).
2. Download your SAP's server certificate from the STRUST transaction. Assuming you setup your SAP server correctly, this should be located in the **SNC SAPCryptolib** folder.
3. If nothing exists under SNC SAPCryptolib, right click on the folder and select **Create** to create a new PSE under **SNC SAPCryptolib**.
4. Open the PSE.
5. Enter a password if prompted. (If not, use the **Password** button to set a password).
6. Click the SNC SAPCryptolib folder.
7. Double click the **Subject** under **Own Certificate**.
8. Confirm the certificate details appear in the **Certificate** section.
9. Click the **Export Certificate** icon button at the bottom to open a dialog box, which allows you to download the certificate.

> If the button is not enabled, you may need to click the Display or Edit (pencil and glasses) button and click the Base64 selection button when prompted and then the green checkmark button to complete the download.

Personal Security Environment Setup

As with your SAP server setup, you should consult the latest SAP documentation for more information when setting up your Personal Security Environment (PSE). These instructions are provided to illustrate the options to configure the SAP SNC Account secret template in Secret Server, but SAP's documentation may provide more information about your options pertaining to the creation of a PSE. To set up your PSE:

1. In your client environment (your Secret Server or distributed engine server), create a directory to stage your setup. For example, I used `C:\SAPSNC`.

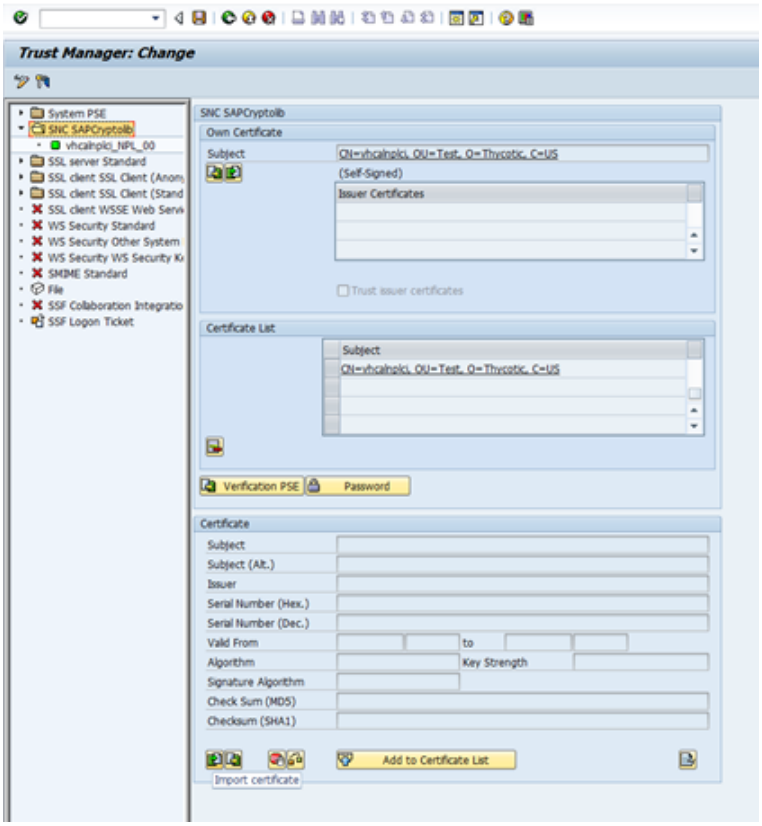
2. Add the two SAP NCO files (sapnco.dll and sapnco_utils.dll), the SAP Cryptographic library (sapcrypto.dll), your ticket license file, and sapgenpse.exe to this directory.
3. Copy the server certificate you exported from your SAP instance to this directory.
4. Add two system environment variables to your server:
   - SECUDIR should be the directory you just created (for instance `C:\SAPSNC`)
   - SNC_LIB should be the full path of the SAP Encryption library (for instance `C:\SAPSNC\sapcrypto.dll`).
5. Following SAP's instructions, use SAPGENPSE (or other tools that SAP may provide) to generate the PSE, including the cred_v2 file and the X.509 certificate. See Configuring the Use of the SAP Cryptographic Library for SNC. For example, you could run these commands from a command prompt window with Administrator permissions in the `C:\SAPSNC` directory:

   ```
   sapgenpse get_pse -p target.pse -x <PASSWORD> <DISTINGUISHED NAME>
   sapgenpse seclogin -p target.pse -x <PASSWORD> -O <DOMAIN\USER>
   sapgenpse maintain_pk -a <CERT FILE FROM SAP GUI> -p target.pse -x <PASSWORD>
   sapgenpse maintain_pk -v -l -p target.pse -x <PASSWORD>
   sapgenpse export_own_cert -o target.crt -p target.pse -x <PASSWORD>
   ```
6. When you create the server credentials with the sapgenpse "seclogin" command, specify a Windows or Active Directory user for the credentials. You have two options here:
   - Specify the same user who runs your Secret Server or distributed engine as the one who is allowed to use the PSE you just setup. This is the easier option. Secret Server
   - Specify a different Windows or Active Directory user. If you choose this option, you need to also create a secret for that user in Secret Server as either a Windows or Active Directory secret. Add this secret to your SAP SNC secret's associated secrets.

For more information about using the SAPGENPSE tool, see Creating the Server's Credentials Using SAPGENPSE.

Importing PSE information to the SAP GUI

1. As above, refer to SAP's documentation for details on getting your PSE recognized by your SAP server. This is just an example.
2. Import the certificate you created above ('target.crt' in my example) through the STRUST transaction in the SAP GUI:
   a. Go to **STRUST \> SNC SAPCryptolib**.
   b. Click the entry below the SNC SAP Cryptolib folder. In the example below it is vhcalnplci_NPL_00.



c. If prompted for a password, type it and then click the green checkmark button.

d. Click the Import Certificate icon on the far left on the bottom (hover over). The Import Certificate dialog box appears.

e. Type your certificate's file path.

f. Click the green checkmark button. The dialog disappears.

g. Confirm the certificate details are now in the **Certificate** section.

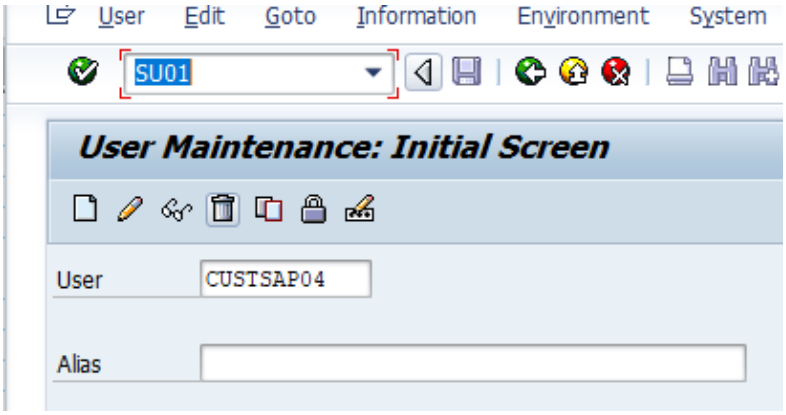
h. Click **Add to Certificate List**.

> If the button is not enabled, you may need to click the Display or Edit (pencil and glasses) button.

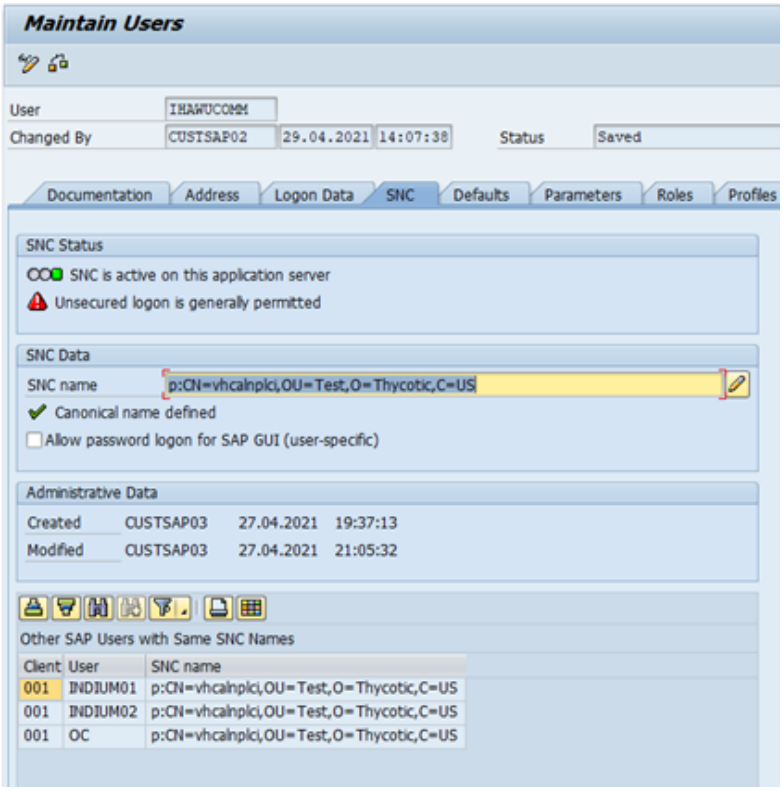
i. Confirm the certificate now appears in the **Certificate List** section.

j. Save and exit.

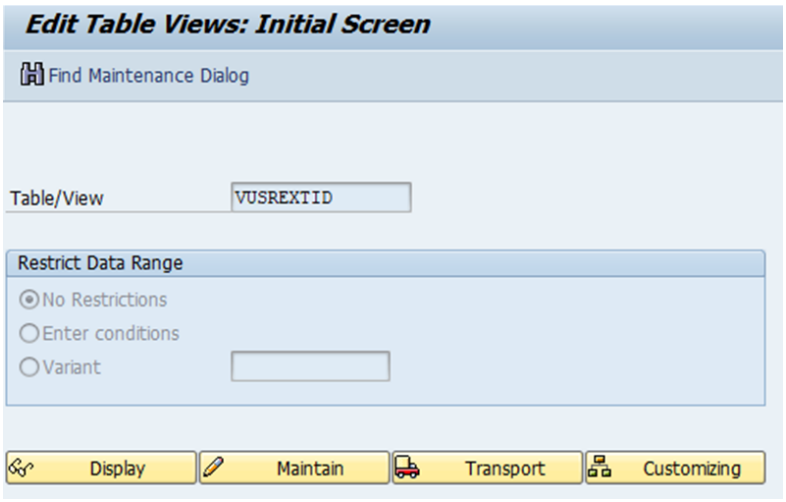
3. Go to the **SU01** function.
4. Type your SAP user's name in the **User** text box.



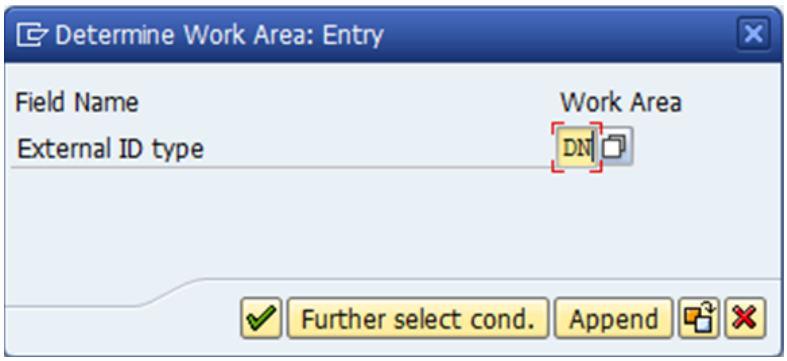
5. Click the pencil icon to edit.
6. In the **SNC** tab, define the SNC name using the syntax: `p:<YOUR USER'S DISTINGUISHED NAME>`.



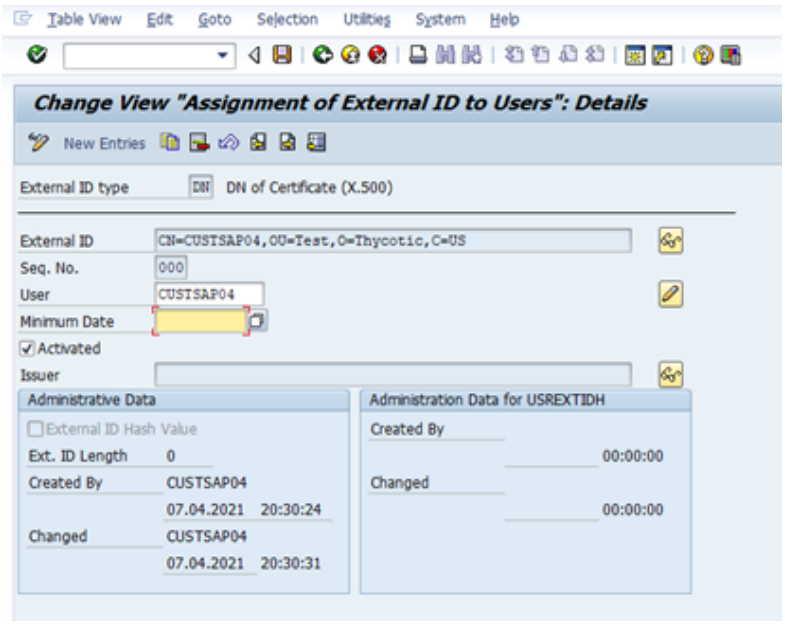
7. Save and exit the **SU01** transaction.
8. Go to the **SM30** transaction.



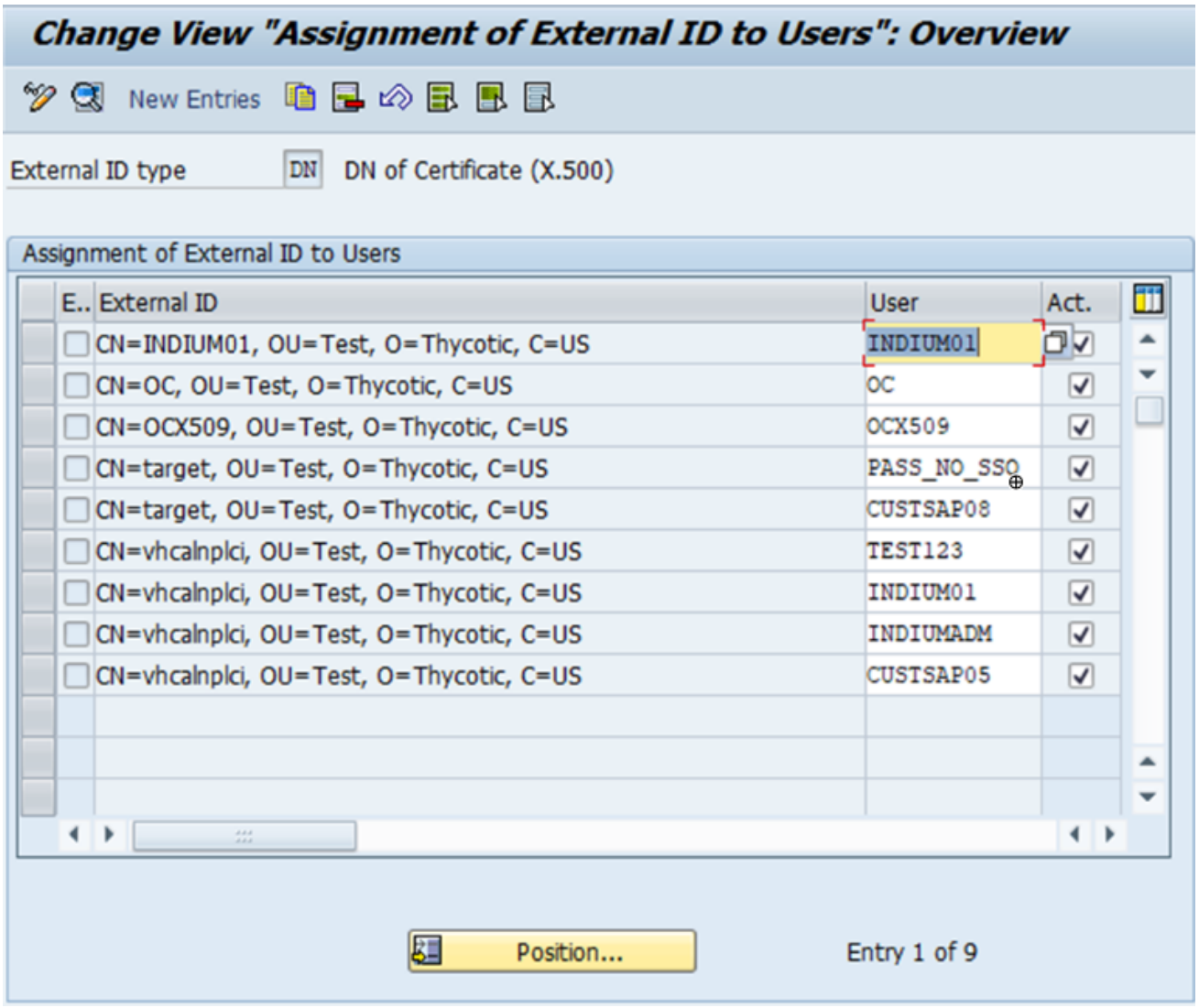
9. Type `VUSREXTID` in the **Table/View** text box.
10. Click the **Maintain** button. A dialog box appears:



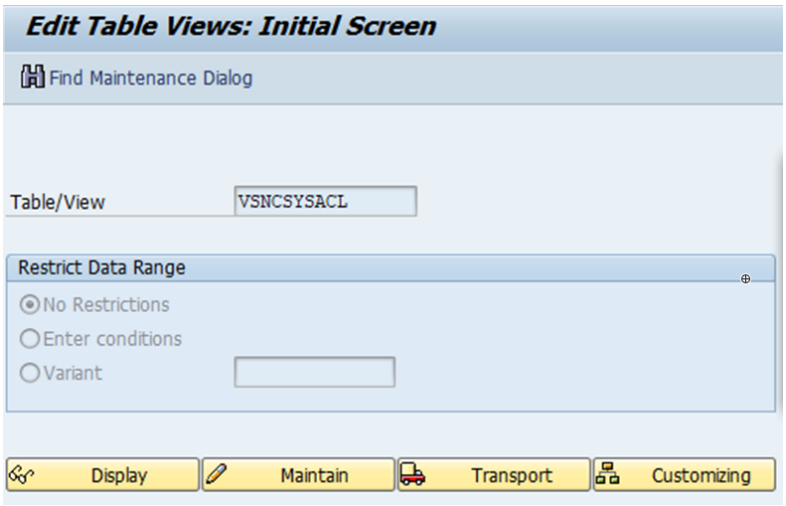
11. Select **DN** as the work area.
12. Click the check mark icon button.



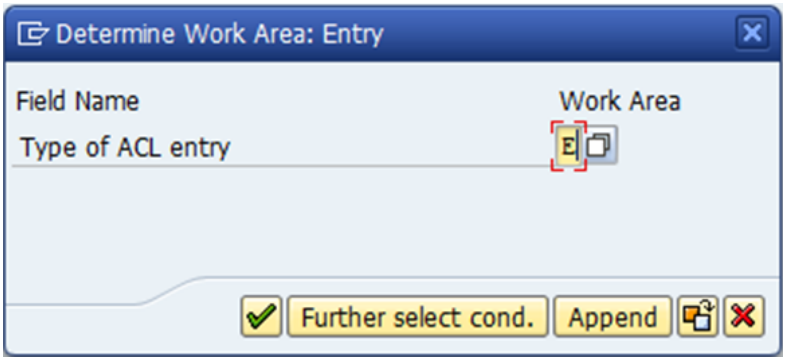
13. Click the **New Entries** button. The Change View "Assignment of External ID to Users" panel appears:



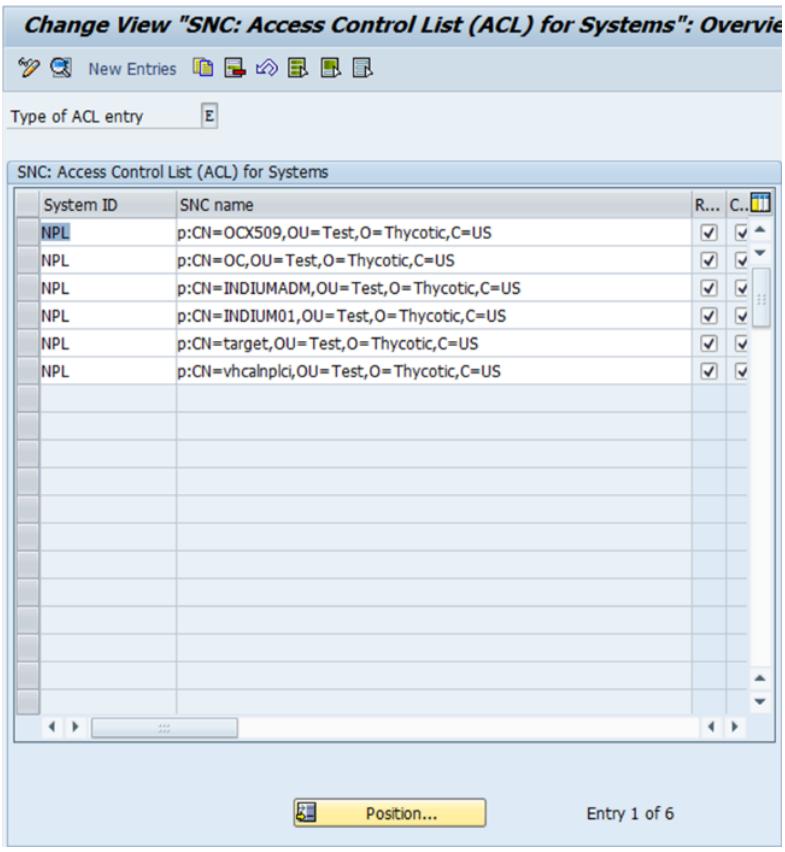
14. Click the Details (magnifying glass) icon. A details panel appears.
15. Fill out the fields as follows:
    a. Replace the **External ID** with your own
    b. Click to select the **Activated** check box.
    c. Type your SAP username in the User text box.
    d. Type a sequence number in the **Seq. No.** (sequence number) text box. For example, 000.
    e. Save and exit.
16. Return to the SM30 function.



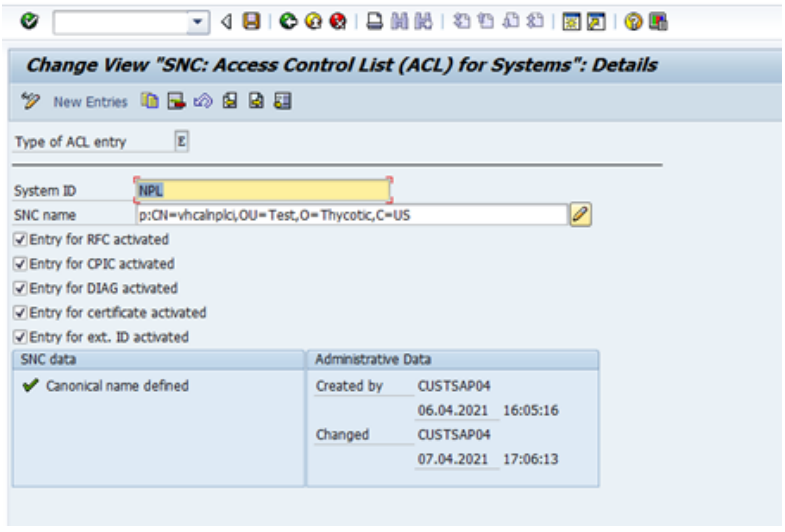
17. Type VSNCSYSACL in the **Table/View** text box.
18. Click the Maintain button. A dialog box appears:



19. Select **E** as the work area.
20. Click the check mark icon button. The dialog box disappears.
21. Click the **New Entries** button. The Change View "SNC: Access Control List (ACL) for Systems" panel appears:



22. Click the Details (magnifying glass) icon. A details panel appears.



23. The **System ID** should match the system ID of your SAP instance. The **SNC name** should be the distinguished name of the server. There should only be one entry in this table for the server.
24. Confirm that a "Canonical name defined" message appears.

25. Save and exit

Creating an SAP SNC Secret in Secret Server

SAP SNC Account secrets are created in the same way as the original SAP Account secrets but have additional fields, as described above. For details that apply to both the SAP Account and SAP SNC Account secrets, see "SAP Heartbeat and Password Changing" on page 1026.

If your PSE was created for a Windows or Active Directory user other than the one who runs Secret Server or distributed engine, you need to add that user to your SAP SNC Account secret's associated users. To do this, add your user as either a Windows Account or an Active Directory secret. Next, open your SAP SNC Account secret and navigate to the Remote Password Changing tab to add that secret as an Associated Secret

If you do not use single sign-on or if you choose to use the username and password without the X.509 certificate for authentication, the X.509 certificate may be omitted.

Troubleshooting

SAP Account Secret Work but SAP SNC Secrets Do Not

SNC uses port 4800 to communicate. If the original SAP Account secrets work but SAP SNC secrets do not, be sure that port 4800 is not blocked by your firewall or VPN.

Client-Side Errors

If you experience client-side errors (such as generating a client certificate), right click on your SAP DLL files (`sapcrypto.dll`, `sapnco.dll`, or `sapnco_utils.dll`), and make sure that they are not blocked by your OS.

Distinguished Name Errors

If you run into an error message a distinguished name (DN) error, such as `Exception: LOCATION CPIC (TCP/IP) with Unicode ERROR GSS-API(maj): No credentials were supplied Unable to establish the security context target="p:CN=vhcalnplci, OU=Test, O=Thycotic`, check your spacing in the distinguished name. SAP can be strict about adding or removing the spaces after commas in the DN.

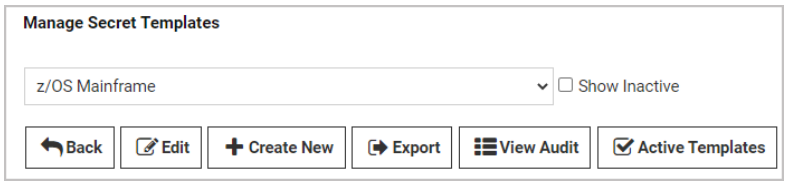
Create and Customize an IBM iSystem (AS/400) Template to use the new IBM iSeries (AS/400) Password Changer

The IBM iSeries (AS/400) Terminal password changer is based on the z/OS Mainframe password changer. It uses the 5250 terminal connection and scripting to perform the password change and heartbeat. You can modify the script for any advanced configuration requirements, and Delinea Professional Services is available to help you.

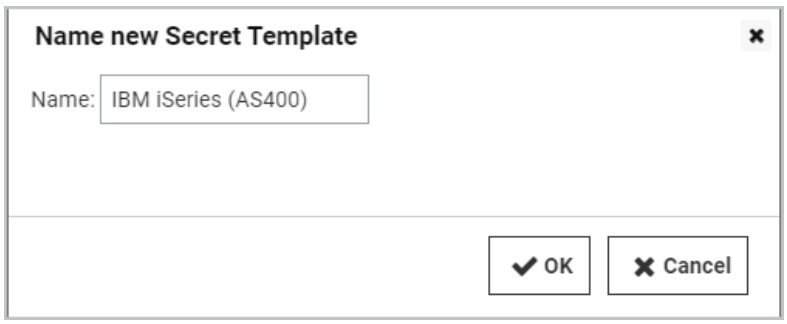
You can also change passwords on the AS/400 using SSH. See "Creating a Custom Password Changer for IBM AS/400" on page 1007.

Create an AS/400 Secret Template

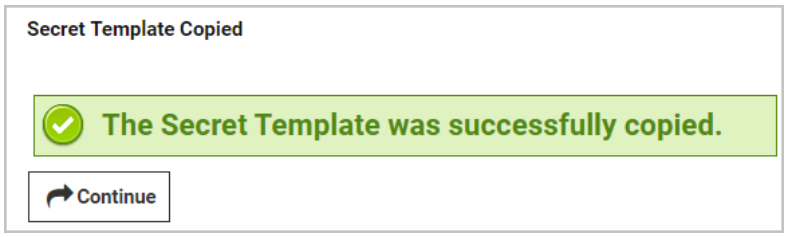
1. Navigate to **Admin > Secret Templates**.
2. On the **Manage Secret Templates** page, select the **z/OS Mainframe** template from the drop-down list.
3. Click the **Edit** button.



4. On the **Secret Template Designer** page, click the **Copy Secret Template** button.
5. On the popup page, type `IBM iSeries (AS400)` in the **Name** text box.



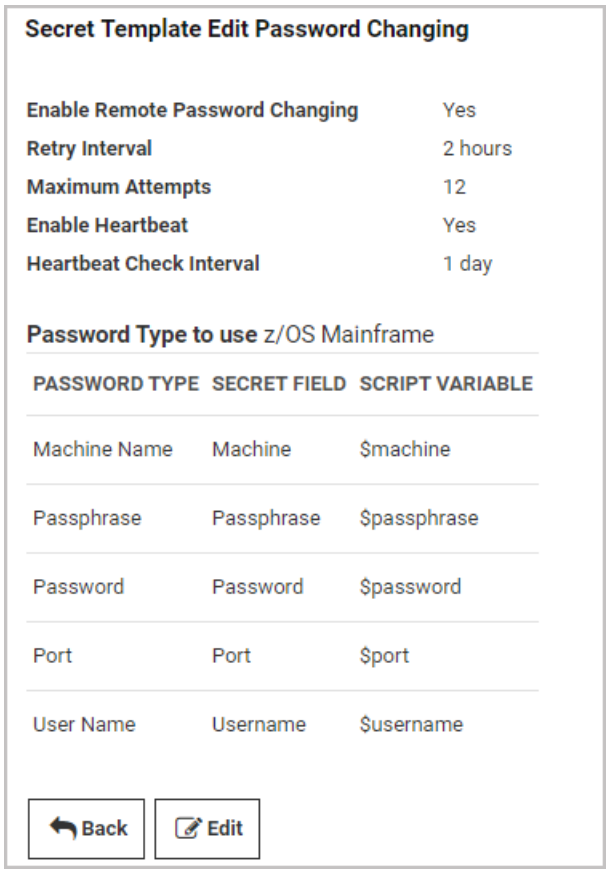
6. Click the **OK** button.
7. On the confirmation page, click the **Continue** button.



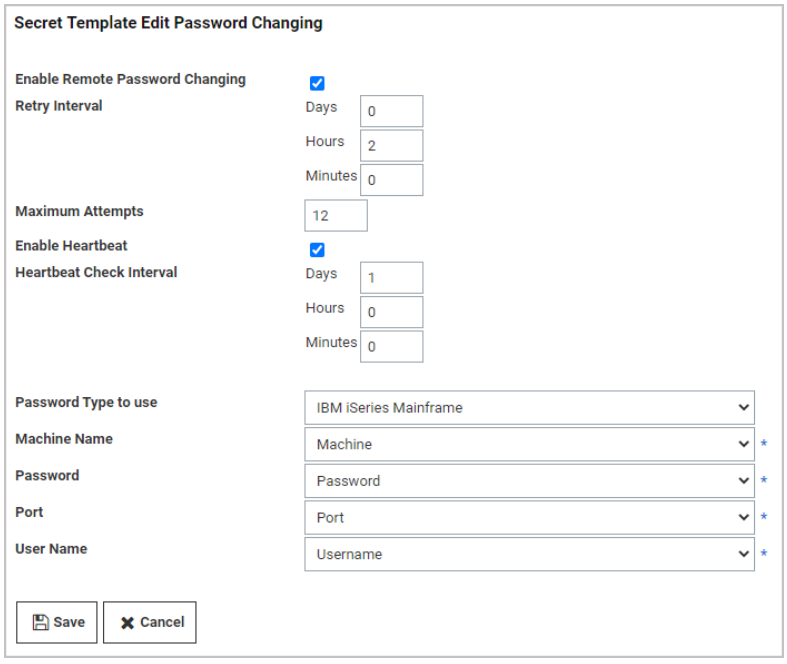
Optional: on the **Secret Template Designer** page, you can deactivate the **Passphrase** field by clicking the deactivate icon to the right of the **Passphrase** row. Unlike the z/OS, the iSeries does not need an additional passphrase and will not have an option for it unless adjusted. Unless your environment specifically requires the passphrase text-entry field, we recommend deactivating it.

Modify Your AS/400 Secret Template to use the AS/400 Password Changer

1. On the **Secret Template Designer** page, click the **Configure Password Changing** button.
2. On the **Secret Template Edit Password Changing** page, click the **Edit** button. The page becomes editable.



3. Next to **Password Type to Use**, click the drop-down list and select **IBM iSeries Mainframe**.

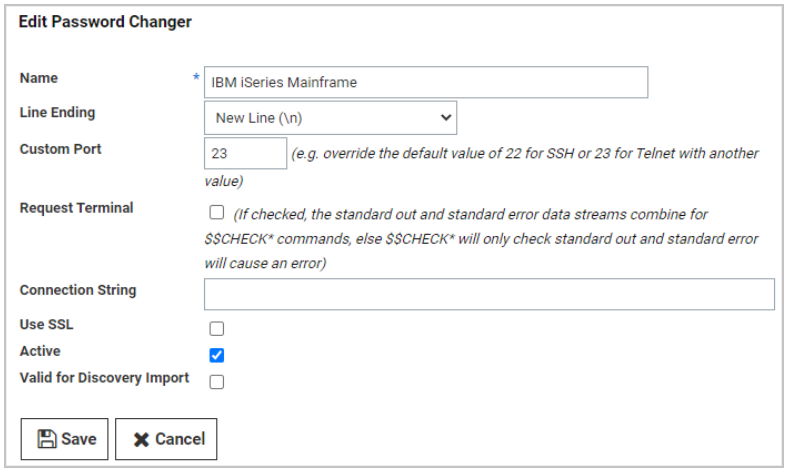


4. Make required changes, if any, to the text boxes and lists.
5. Click the **Save** button. The page is no longer editable.
6. Click the **Back** button.
7. On the **Secret Template Designer** page, create secrets based on the new template as desired.

Customize Your AS/400 Password Changer for Your Environment

For the default IBM iSeries (AS/400) systems, the default password changer configuration requires no adjustment. However, additional parameters and connection string options are available.

1. Navigate to **Admin > Remote Password Changing**.
2. Click the **Configure Password Changers** button.
3. On the **Password Changer Configuration** page, click the **IBM iSeries Mainframe** link.
4. On the **IBM iSeries Mainframe** page, scroll to the bottom and click **Edit**.
5. On the **Edit Password Changer** page, adjust ports and other parameters as desired.



6. Click the **Save** button.

The *trace* function can be a powerful tool for troubleshooting and debugging, especially for complex RPC implementations in unique environments. The trace function logs emulator input, mainframe output, and ASCII screenshots of what is happening on the terminal GUI. To write a trace file to the Secret Server website or engine, just add TRACEto the connection string. If using the model option, this needs to be an integer, for example, model=4.

It is important to delete trace files after debugging–they could contain sensitive data.

Additional Functions, Adjustments, and Parameters

For unique IBM iSeries environments, the IBM iSeries password changer offers extra features, options, adjustments and parameters for customization, including the commands in the table below. To implement these commands successfully, it helps to keep in mind that the password changer is emulating user input. Some of these commands are designed for very fine emulations of unique IBM iSeries environments, and Delinea Professional Services can help you with these. Other commands are implemented and tested on a base environment, so before implementing them in a production environment, you should verify that they are working as expected through testing or by using the trace function.

The commands below are followed by an <ENTER> command by default. To prevent this, you must add ##NOENTER in the *comment* of the previous command. For example:

ORDER	COMMAND	COMMENT	PAUSE(MS)
1	$USERNAME	Username ##NOENTER	2000
2	<TAB>	Tab to password	2000
3	$CURRENTPASSWORD	Password	2000

Table: Secret Template Commands

Command	Action	Description or Example
<Backtab>	Tab to the previous input field.	
<Clear>	Clear the screen.	Mostly used for trace.
<Close>	End the session to the mainframe.	
<Delete>	Delete a character under the cursor; can be used with <MoveCursor(#, #)>	
<DeleteField>	Delete the entire text input or field.	
<DeleteWord>	Delete the current word if available, otherwise delete the previous word.	
<Disconnect>	Disconnect the password changer's connection to the mainframe.	
<Down>	Move cursor down.	
<Enter>	Send the Enter key press command.	

Command	Action	Description or Example
<Erase>	Erase previous character on a selected text input.	<Erase>
<EraseEOF>	Erase end-of-field of current text input.	<EraseEOF>
<Execute( )>	Execute commands in shell.	<Execute(USRMGR)>
<HexString(#)>	Insert a control character in a text field or string.	<HexString(41)>
<Key(#)>	Execute named iSeries keys.	Execute unique keys via hex, character code, or key symbol.
<Left>	Move cursor left.	
<PF(#)>	Execute program function.	Program function keys 1 to 24
<PA(#)>	Execute program attention.	Program attention functions 1 to 3
<MoveCursor(#, #)>	Move the cursor by row and column.	<MoveCursor(10,2)>
<Right>	Move cursor right.	
<Tab>	Tab to the next line.	
<Up>	Move cursor up.	

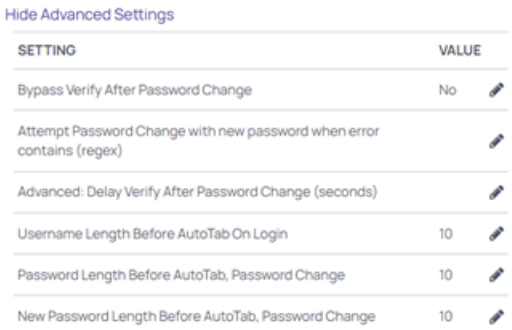
IBM iSeries Mainframe

Usernames and passwords can run into length issues during remote password changing and heartbeat. The issue stems from the behavior in the client; when a user logs in and enters a username of 10 characters, the client will auto-tab to the next field (the password field). The heartbeat and remote password changing process automatically inserts this tab, which can cause improper behavior in the headless client when the username or password is 10 characters.

This can be avoided by setting three properties on the Custom Commands for the IBM iSeries Mainframe password changer:

- Username Length Before AutoTab on Login
- Password Length Before AutoTab, Password Change
- New Password Length Before AutoTab, Password Change

Go to **Remote Password Changing** > **Configure Password Changers** > **IBM iSeries Mainframe**.



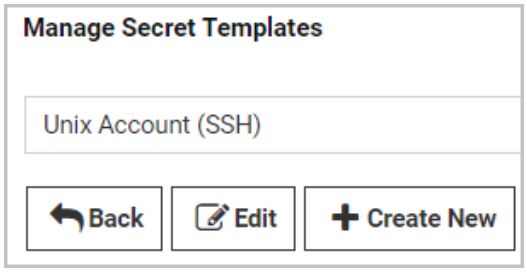
Typically, the console will auto-tab after 10 characters. If your environment behaves differently, note in the console how many characters are entered until auto-tab to the next field occurs, and enter that number into the proper field.

Creating a Unix Account Secret Template that Uses Key Authentication Instead of a Password

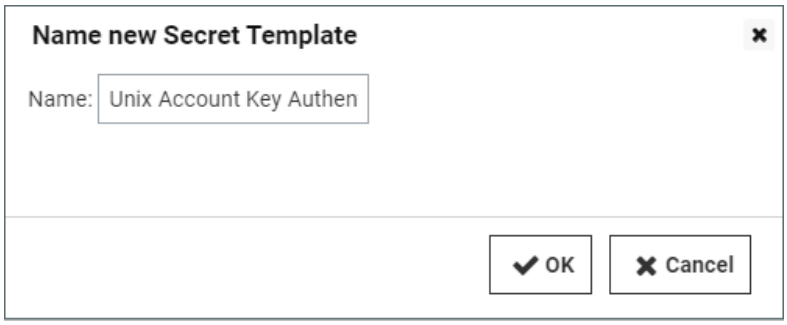
To create a Unix account secret template that uses key authentication only instead of a password, begin by using an existing **Unix Account (SSH)** template as a baseline.

Create the New Template

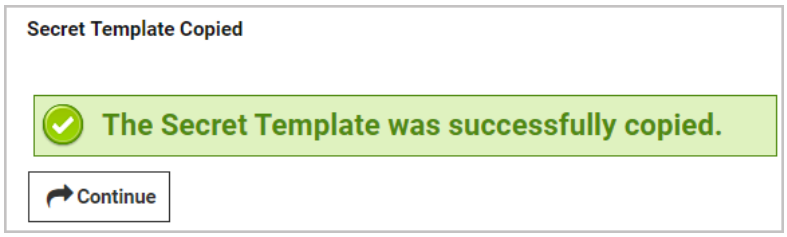
1. Go to **Admin > Secret Templates**.
2. Select the built-in **Unix Account (SSH)** template from the drop-down menu and click **Edit**.



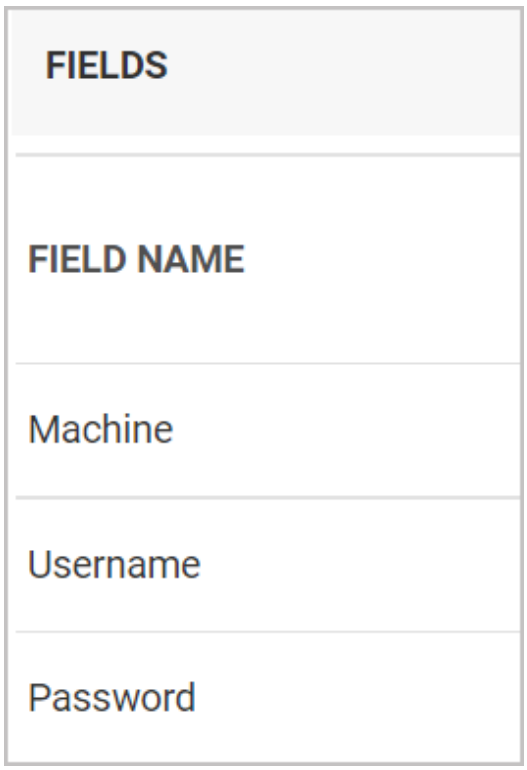
3. On the **Secret Template Designer** page, scroll to the bottom and click **Copy Secret Template**.
4. Give the new template an appropriate name, such as *Unix Account Without Password (SSH)* or *Unix Account Key Authentication Only (SSH)*.



5. Click **OK**.
6. On the **Secret Template Copied** confirmation page, click **Continue**.



7. On the **Secret Template Designer** page, scroll down to the **Fields** section and under **Field Name**, find the **Password** row.



8. At the right end of the **Password** row, click the **Edit this field** icon
9. In the **Password** row under **IS REQUIRED**, uncheck the box. Optionally, you can also select **Not Editable** from the **Edit Requires** drop-down list.

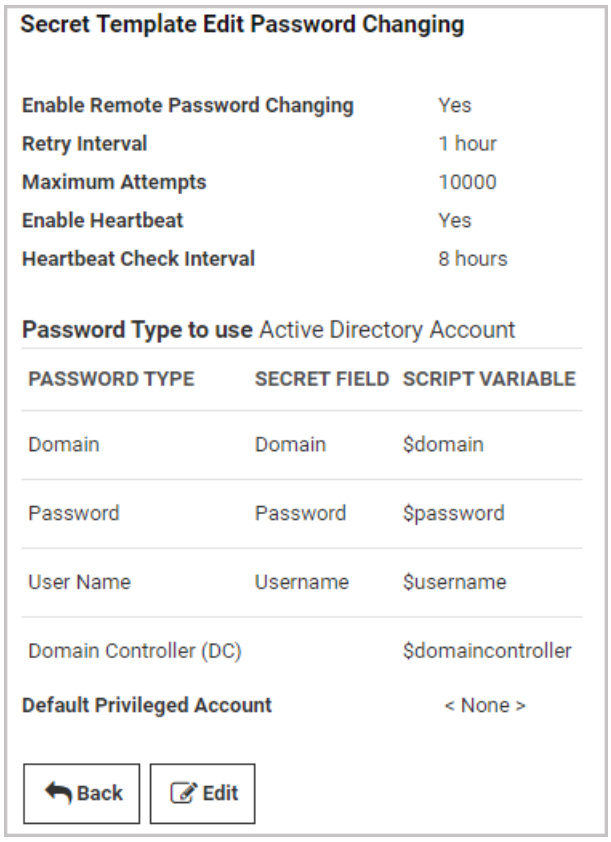
10. At the right end of the **Password** field, click the **Save this field** icon

You now have a Unix account (SSH) Secret template that displays key authentication fields instead of a password field.

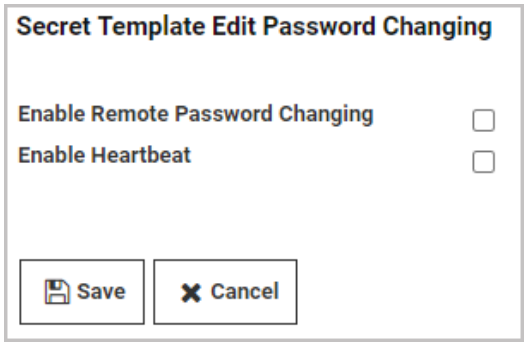
Disable Remote Password Changing and Heartbeat

Your new template has inherited characteristics from the **Unix Account (SSH)** template you based it on, including having **Remote Password Changing** and **Heartbeat** enabled by default. But because your new template has no password, it cannot be remotely changed and heartbeat cannot validate on an empty password. Therefore, you must disable these features by editing your new template using the procedure below:

1. In the **Secret Template Designer** window, scroll to the bottom and click **Configure Password Changing**.
2. In the **Secret Template Edit Password Changing** window, click **Edit**.



3. In the next **Secret Template Edit Password Changing** window:
   a. Uncheck **Enable Remote Password Changing**.
   b. Uncheck **Enable Heartbeat**.



4. Click **Save**.

> Some Secrets based on the **Unix Account (SSH)** might display the **Password** field as well as the **Private Key** and **Private Key Passphrase** (key authentication) fields. If a user signs in using this template with correct credentials in the key authentication fields but a blank or incorrect password in the Password field, the default PuTTY launcher will use key authentication to connect.

Configuring Oracle Secret Templates

Secret Server now has four Oracle templates, three current and one legacy:

- Oracle Account
- Oracle Account (TCPS)
- Oracle Account (Template Ver 2)
- Oracle Account (Walletless)

Introduction

Overview

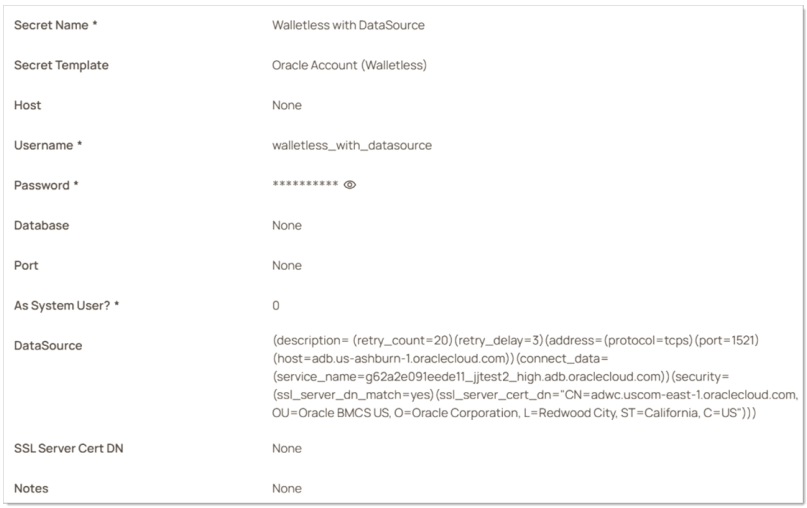
Secret Server now has three secret templates based off the Oracle Managed Data Access NuGet library. Unlike earlier Oracle secret templates, templates, using the NuGet library does not require `Oracle.ManagedDataAccess.dll` to be installed alongside Secret Server or its engines. Additionally, two of these templates support Oracle's TCPS connection protocol. You can run the new templates alongside the earlier Oracle secret templates, but we recommend using the new templates when creating new Oracle secrets.

DataSource Field

All three new templates include an optional DataSource field. The DataSource field acts like a connection string to the Oracle database. When used, it is not necessary to fill out the Host, Database, Port, or SSL Server Cert DN fields. On the secret template's page, "none" appears in each of those fields.

Without DataSource:

With DataSource:



Secret Name *	Walletless with DataSource
Secret Template	Oracle Account (Walletless)
Host	None
Username *	walletless_with_datasource
Password *	**********
Database	None
Port	None
As System User? *	0
DataSource	(description= (retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1521)(host=adb.us-ashburn-1.oraclecloud.com))(connect_data=(service_name=g62a2e091eede11_jjtest2_high.adb.oraclecloud.com))(security=(ssl_server_dn_match=yes)(ssl_server_cert_dn="CN=adwc.uscom-east-1.oraclecloud.com, OU=Oracle BMCS US, O=Oracle Corporation, L=Redwood City, ST=California, C=US")))
SSL Server Cert DN	None
Notes	None

As System User Field

All new templates also have the “As System User” field. It allows connections to the database as the system user. Setting this to “1” allows database connections with the SYSDBA privilege. Setting this to “0” connects with the default role.

Templates

Oracle Account

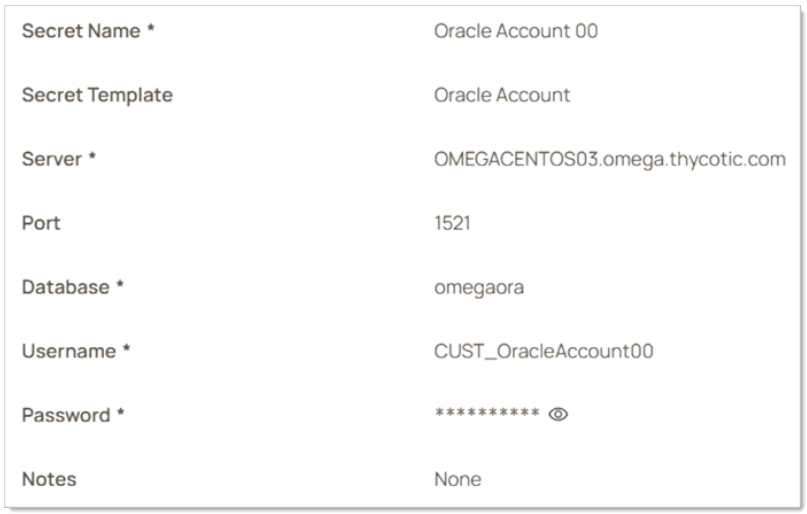
This is the deprecated original template that maintained for legacy implementations. I requires that `Oracle.ManagedDataAccess.dll` is installed alongside Secret Server or its engines.

Oracle Account (Template Ver 2)

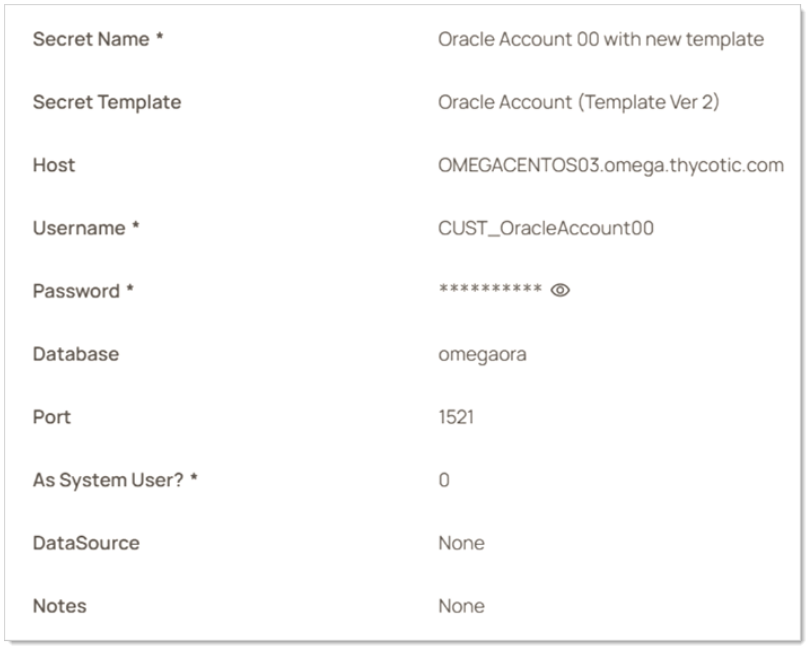
Oracle Account (Template Ver 2) is the closest equivalent to the original Oracle Account template. It does not support Oracle TCPS, but it does use the Oracle Managed Data Access NuGet library. The fields on this secret template are the same as in the original Oracle Account template with a few exceptions. We added two new fields, the DataSource and “As System User” fields described above. Additionally, the “Server” field is called “Host” in the new template to more closely match the terminology in Oracle’s connection string.

Original template:

Secrets



New template:



See "Oracle Account Secret Template for RPC" on page 1021 for more using this template.

Oracle Account (TCPS)

Overview

You can make TCPS connections using the "Oracle Account (TCPS)" secret template and an Oracle Wallet. As described by Oracle, “Oracle Wallet is a container that stores authentication and signing credentials.”

> See Understanding Oracle Wallet for more information, and refer to the documentation on your specific Oracle database for detains on obtaining and using your wallet.

Wallet Location

Prior to setting up Oracle Account (TCPS) secrets, you will need to place copies of your Wallet files on the same server(s) as your Secret Server site(s) where you will have Oracle Account (TCPS) secrets. Afterwards, use the "Wallet Location" field to note the location of your wallet files on your newly created secret. Note that you will need to ensure that the user running Secret Server's app pool (or engine when applicable) is granted permissions to access the directory where the wallet is located.

TNS Admin

The "TNS Admin" field is optional. If you have a `tnsnames.ora` file, specify its containing directory in this field.

As with wallets, you should consult documentation specific to your Oracle database for information on using `tnsnames.ora`. However, in general, `tnsnames.ora` file is a configuration file containing network service names mapped to connect descriptors (for local naming method) or net service names mapped to listener protocol addresses.

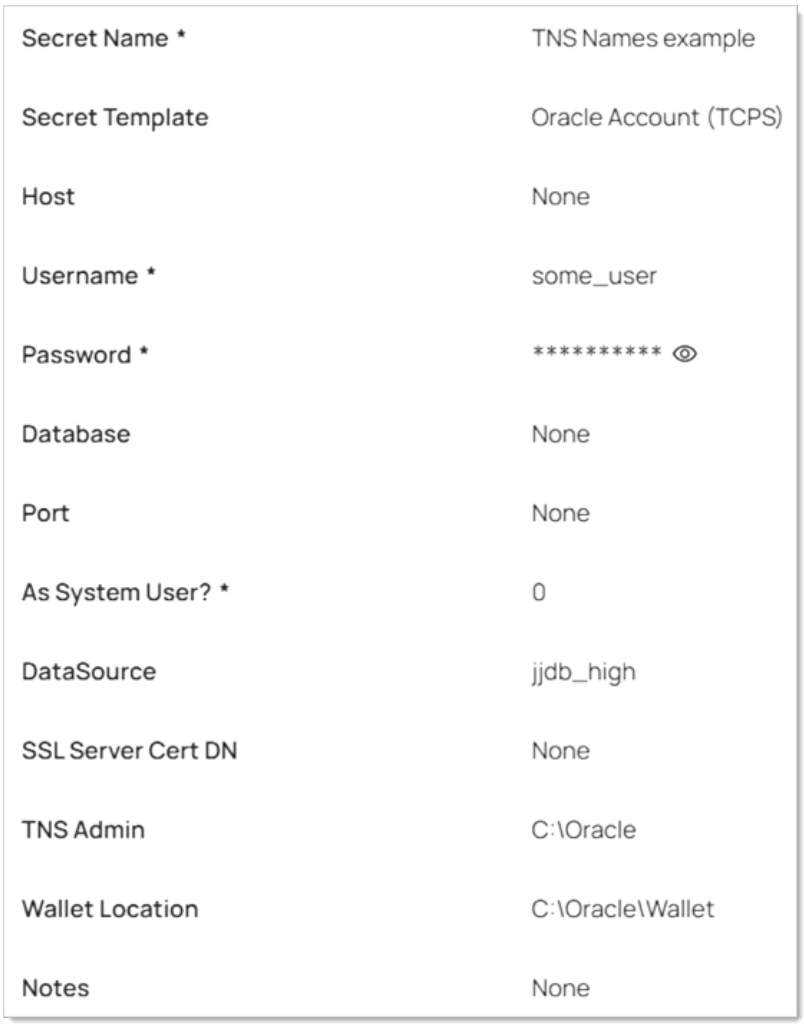
> See Local Naming Parameters in the tnsnames.ora File for more information.

Thus, you can use the `tnsnames.ora` file as an alternative way to specify a connection string. First, put the directory of the `tnsnames.ora` file in the "TNS Admin" field. Second, format the contents of `tnsnames.ora` as `<ALIAS> = <CONNECTION STRING>`. Paste the desired alias from inside the `tnsnames.ora` files in the DataSource field of the secret. For example, if we had a file at `C:\Oracle\tnsnames.ora`, the contents might be as follows:

```
jjdb_high = (description= (retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1522)
(host=adb.us-ashburn-1.oraclecloud.com))(connect_data=(service_name=g62a2e091eede11_jjdb_
high.adb.oraclecloud.com))(security=(ssl_server_cert_dn="CN=adwc.uscom-east-
1.oraclecloud.com, OU=Oracle BMCS US, O=Oracle Corporation, L=Redwood City, ST=California,
C=US")))

jjdb_low = (description= (retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1522)
(host=adb.us-ashburn-1.oraclecloud.com))(connect_data=(service_name=g62a2e091eede11_jjdb_
low.adb.oraclecloud.com))(security=(ssl_server_cert_dn="CN=adwc.uscom-east-1.oraclecloud.com,
OU=Oracle BMCS US, O=Oracle Corporation, L=Redwood City, ST=California, C=US")))
```

In this example, `jjdb_high` or `jjdb_low` are both aliases. We would choose one, say `jjdb_high`, and paste it into the DataSource field on my secret. I would include `C:\Oracle` in the TNS Admin field. My secret would appear as in the following screenshot:

Secrets



Oracle Account (Walletless)

Oracle has recently announced its support of TCPS without requiring the use of wallets. See Securely Connecting to Autonomous DB Without a Wallet (Using TLS) for details.

Because walletless connections utilize TLS instead of “mutual TLS,” this is potentially less secure than a wallet-based authentication, so you should do your own research before deciding if this is the right approach for you. The advantage of the walletless template is that it allows a secure TCPS connection with the simplicity of the “Oracle Account (Template Ver 2)” template. No wallet files need to be deployed to a server.

The fields on this template are the same as Oracle Account (Template Ver 2) with the addition of the “SSL Server Cert DN” field, which can be found in the TLS connection string. Refer to documentation on your specific Oracle database about how to enable walletless connections, as well as how to obtain the TLS connection string to fill in fields on this secret template.

Privileged Password Security Policy Template

To use the privileged password security policy template, follow the steps below:

1. Download the privileged password security policy template.
2. Open the template as a Microsoft Word document.
3. Remove the "About this Template" and "Customizing the Template" instructions and other author comments.
4. Replace the term "Company X" with the name of your organization.
5. Replace the current logo or add your company logo in the upper left corner.
6. Update all of the company-specific contact information (highlighted yellow).
7. Update the effective date.
8. Revise any policy guidelines to meet your organization's policies.
9. Revise the Violations section to meet your organization's policies.
10. Save your changes.
11. Obtain your management and auditors' approval of the completed policy.
12. Distribute the policy according to your management guidance.

SSH Authentication Templates

Overview

With this Secret Server feature, admins can use private SSH keys for PuTTY launcher sessions as well as for RPC tasks (configurable through password changer settings) and Unix and Linux discovery. Passphrases can additionally be stored, if necessary, to decrypt the private keys for additional security. The Unix Account (SSH) secret template includes text-entry fields for the private key and passphrase by default.

The SSH Key template is included by default and can be used to store SSH keys that can later be selected for use in RPC, discovery or launcher authentication for other secrets:


Starting with version 10.1.000000, Secret Server also supports SSH key rotation on secrets.

The **Unix Account (SSH Key Rotation)** and **Unix Privileged Account (SSH Key Rotation)** secret templates use password changers that change the public key in the account's `authorized_keys` file as well as change the password on the account. Secret Server ships with a password changer and custom command sets that allow an account to change its own public key and password, and a password changer and custom command sets that changes a user's public key and password using a privileged account. These scripts can be customized for different Unix environments.

Settings

Typical settings for these templates include:

- **Secret Template Name**
- **Name Pattern:** You can use a naming pattern to enforce a standardized name for this Secret Template. The naming pattern uses regular expressions.
- **Description**

- **All History:** Save all history of secret names using this template.
- **Secret Name History Length:** Number of secret names to keep in history for this template.
- **Validate Password Requirements On Create:** If enabled, password fields must meet the password requirements when a secret is created.
- **Validate Password Requirements On Edit:** If enabled, password fields must meet the password requirements when a secret is edited.
- **SSH Key Format:** PuTTY or OpenSSH. PuTTY's PPK format is proprietary and mainly used within the PuTTY suite, primarily on Windows systems. OpenSSH's key format, being open and standardized, enjoys broader support and compatibility across different platforms and SSH clients.
- **Required Permission To Edit Password Change Configuration:** Specifies which permission is allowed to edit the password change configuration on a secret from this template.
- **SSH Key Algorithm:** ECDSA (strongly recommended) or RSA (for backwards compatibility). See "ECDSA Versus RSA" below for details.
- **SSH Key Bit Size (RSA only):** 1024, 2048, or 4096 bits.

ECDSA Versus RSA

ECDSA (Elliptic Curve Digital Signature Algorithm) and RSA (Rivest-Shamir-Adleman) are both cryptographic algorithms used for securing data, but they operate on different mathematical principles and offer distinct characteristics, especially when employed as SSH key algorithms:

- **Security:** Both RSA and ECDSA provide high levels of security, but they achieve this through different means. RSA's security is based on the factorization problem of large integers, while ECDSA's security relies on the elliptic curve discrete logarithm problem. Generally, ECDSA can achieve comparable security to RSA with a much shorter key length. For instance, a 256-bit ECDSA key is roughly equivalent in security to a 3072-bit RSA key.
- **Performance:** ECDSA keys are typically smaller than RSA keys for a comparable security level, which means they can be faster for generating and verifying signatures due to the reduced computational complexity. This can lead to quicker SSH key exchanges and potentially faster connections, especially important in environments with a high volume of SSH traffic.
- **Compatibility:** RSA is more universally supported across different systems and SSH implementations due to its longer presence in the market. ECDSA support is widespread but slightly less universal, which could affect interoperability in diverse environments.
- **Key Length and Scalability:** RSA keys usually start at 2048 bits for adequate security, with 3072 and 4096 bits being common for increased security. ECDSA keys, due to their efficiency, start at 256 bits (equivalent to P-256 curve), with P-384 and P-521 being options for higher security needs. This makes ECDSA more scalable and efficient as security demands increase.

In summary, ECDSA offers a more efficient and potentially faster alternative to RSA for SSH key algorithms, especially where high security with lower computational overhead is desired. However, RSA's universal compatibility and long-standing reputation may make it a preferable choice in environments prioritizing interoperability and proven security mechanisms.

Secret Server Cloud

Secret Server Cloud is a scalable, multi-tenant cloud platform that offers the same robust features as the on-premise Secret Server Professional edition. Hosted on the Microsoft Azure platform, Secret Server Cloud ensures that all backend services, databases, and redundancy are securely managed by Delinea, eliminating the need for customers to manage these components directly. This cloud-native solution provides enterprise-grade Privileged Access Management (PAM) capabilities, including secure vaulting of privileged credentials, automated password management, and comprehensive auditing and reporting tools. Secret Server Cloud is designed to help organizations quickly secure and manage their privileged accounts, ensuring compliance and reducing the risk of cyber attacks.

Secret Server Cloud Quick Start

Overview

Secret Server is a scalable, multi-tenant cloud platform that provides the same features as the on-premise Secret Server Professional edition. With Secret Server Cloud, all backend services, databases, and redundancy are securely managed by Delinea and hosted on the Microsoft Azure platform. Customers do not have direct access to the databases or application file system.

End users are also referred to as "business users."

Cloud Versus On-Premise Secret Server

For documentation purposes, Secret Server Cloud is the same as the corresponding on-premise edition. However, there are some feature differences:

- **Site Connectors:** On-premise versions can use multiple site connectors to manage engine connections, such as RabbitMQ or MemoryMQ. The cloud version manages this for you as an Azure service and is not configurable.
- **CRM Integration:** On-premise versions can integrate with CRMs via direct database connections or the ConnectWise API. This is not currently available in Secret Server Cloud.

Getting Started

This section walks you through an initial configuration of your cloud instance. To see additional documentation for Secret Server Cloud features, please refer to the support resources section at the end of this document.

System Requirements

All cores are physical unless otherwise noted.

A distributed Engine server is required to communicate with Secret Server Cloud. Distributed engine server recommended specifications:

A distributed engine server is required to communicate with SSC. Distributed engine server recommended specifications:

- Windows Server 2016 or Above
- CPU: 4-core 2 GHz (minimum)
- Memory: 4 GB of RAM (minimum)

Engine Connectivity

Secret Server Cloud's Architecture Diagram shows the network topology of your cloud instance. Your on-premises distributed engines do not need any inbound TCP/IP ports open (unless using RADIUS authentication). If you do not have outbound firewall policies in place, no firewall configuration is necessary. If you do, the distributed engines need outbound access to:

- Secret Server Cloud's multi-tenant front-end Web server
- A shared service bus
- A customer-specific service bus
- A Content Delivery Network (CDN)

The protocols and endpoint details are in the architecture diagram mentioned above.

Initial Setup

After you sign up for a trial, you can choose your URL name and provision your instance:

> To see additional documentation for Secret Server Cloud features, please refer to the support resources section at the end of this document.

1. After you sign signed up for a Secret Server Cloud trial, you received an email from Delinea Sales. Click the **Cloud Portal** link in that email to begin your setup. The Setup Page appears in your browser.
2. Choose your location in the **Cloud Environment** dropdown list.
3. Click the **Continue** button. The Delinea One Portal appears.
4. Create the password for your first user account with administrator credentials. This account will be assigned to the email address you entered to request the trial.
5. After confirming the password, click the **Set Password and Login** button. The Delinea log on page appears.

> This is the backup admin account that you may need in a "break the glass" or unlimited admin situation. Delinea recommends you store the password in a secured physical location such as a safe or locked file cabinet. You can reset the password using an email reset, but **if this password is forgotten or you no longer have access to the email account, Delinea cannot reset this password**.

6. Click the blue button that matches the location you just chose. A setup page appears.
7. Type a name for your subdomain. Do not use special characters or spaces.
8. Read the Business User (End User) License Agreement.
9. Click to select the check box to signify agreement.

10. From the dropdown, select **Yes** or **No** to signify your organization's oversight of EU information.
11. Click the **Accept** button. It may take several minutes for your new Secret Server Cloud to spin up.
12. When initialization is complete, click go to your Secret Server Cloud URL and click the **Login with Delinea One** button. You are automatically redirected to your new Secret Server Cloud dashboard.

> For information on how to install a distributed engine, please refer to "Distributed Engine Installation " on page 763.

Configure Active Directory Integration

Active Directory integration allows users to log in with their domain credentials. Connections to your domain are routed through the distributed engine service running in your network.

1. On the dashboard, create a new Active Directory secret from the create secret widget in the upper right hand corner.

   > The domain account should be able to read users and groups from the domain you want to sync. For detailed information on the rights required, please see the "Active Directory Rights for Synchronization Account" on page 493.

2. Type the domain, username, and password in the **Create Secret** form.
3. Save the secret.
4. Navigate to **Admin > Active Directory**.
5. Click **Edit** and check the boxes for **Enable Active Directory Integration** and **Enable Synchronization of Active Directory**.
6. Click the **Save** button.
7. Click the **Edit Domains** button.
8. Click the **Create New** button.
9. Type your FQDN and a friendly domain name that users will see on the login page.
10. Click **Sync Secret** to select the secret you just created.

    > The domain site is set to default. This means that the Active Directory authentication and synchronization will run through the distributed engine service installed on your network.

    > Do **not** select "Enable Login from AD." If you do, you cannot set the domain groups later in this instruction.

11. Click the **Save and Validate** button.
12. Click the **Back** button.
13. Click the **Edit Synchronization** button. The Synchronization Edit page appears.

14. In the **Available Groups** list, click each domain group that you want to log on in Secret Server Cloud instance and click the **<** button to move the group to the **Synchronized Groups** list.
15. Click the **Save** button.
16. Click the **Synchronize Now** button to start the user and group synchronization immediately. The synchronization process runs automatically, but to get immediate results, you can start it manually.

Test Heartbeat and Remote Password Changing

Heartbeat ensures the secrets you have stored have the correct password, and Remote Password Changing (RPC) changes passwords on demand or a schedule.

1. Navigate to **Admin > Remote Password Changing**.
2. Click the **Edit** button.
3. Click to select the **Enable Remote Password Changing** and **Enable Heartbeat** check boxes.
4. Click the **Save** button.
5. Click the **Run Now** button in the **Remote Password Changing and Heartbeat Log** sections. This runs the heartbeat and RPC processes immediately.
6. Go to the secret you created for domain synchronization in the previous section or create a new test secret to use.
7. A brand new secret's **Last Heartbeat** status should be pending or processing. Once heartbeat completes you should one of these statuses:
   - **Unable to Connect:**Secret Server could not reach the target machine. This could be a firewall issue or the machine name or IP address is wrong.
   - **Failed:**Secret Server could connect but could not authenticate. This likely means the password on the secret is incorrect.
   - **Success:**Secret Server successfully connected with the username and password.
8. You can test password changing by viewing a secret and clicking the **Change Password Remotely** button.

   
   This will change the password on the target system.
9. You can view the status of password changes and heartbeats in the log at **Admin > Remote Password Changing.**

Next Steps

- Add another user to the Administrator role in Secret Server. This allows you to have another administrator besides the initial user account created. To assign roles, go to **Admin > Roles** and click the **Assign Roles** button.
- Add a folder and share it with the group you synchronized from Active Directory. Create and edit folders from the Folder Tree View on your Dashboard.
- Create a secret in that folder for other users to see. When creating a secret, you can click the **Folder** link to save it to another folder.

- Have other users log on. Any users synchronized to Secret Server through the domain synchronization can log on with their domain credentials.
- Enable Google two-factor authentication by going to **Admin > Users**, editing the specific user, and assigning a two-factor option.

Troubleshooting and Resources

Get Error: "Site (Default) engines are not currently online" When Saving Domain

This can occur when Secret Server was not able to complete a round trip with the installed engine service. This validation may take several minutes for Secret Server to perform after the engine has been approved and assigned to the site. To address the issue:

1. On the server you installed engine on, check the logs in the install directory `C:\Program Files\Thycotic Software Ltd\Distributed Engine\log`.
2. If you see a message for "Could not configure, trying in 30 seconds" or a "Bus Broken Down Error" verify that the engine is approved and assigned to your default site.
3. Go to the site under **Admin > Distributed Engine > Manage Sites**.
4. Click the **Validate Connectivity** button.
5. If a success message appears and the engine status shows as online, try saving the domain again.

Secret Server Character Limits

Secret Server allows the following number of characters for the fields:

- Folder Name: 128. Error Message when exceeded - *The folder name must be 128 characters or less*.
- Group Name: 250. Error Message when exceeded - *Application Error*.
- Role Name: 255. Error message when exceeded - *Please choose a Role name with between 1 and 255 characters*.
- Secret Name: 1,992. Error Message when exceeded - *Invalid Secret Name*.
- Secret Text Fields: 9,999. Error message when exceeded - *Secret item value exceeds max length characters.*
- Secret Note Field: 9,999
- Request field 600 (This is a comment field used when requesting access to a secret). No error message given, it prevents exceeding 600 characters. If you paste in more than that it clips the text and still lets you save.
- Local User Username: 128
- Local User Display Name: 256
- Local User password: 500

AWS Key Management in Secret Server Cloud

> Managing your own encryption key or using a third-party provider, such as AWS KMS, has very serious ramifications if not carefully handled–you can lose access to your Secret Server data. When using AWS KMS, Secret Server requires access to the AWS KMS key for the website to be accessible and secrets to be available. If the AWS KMS key is deleted, Secret Server becomes permanently unable to decrypt any data–all access to secrets is lost. If the credentials that Secret Server uses to access the AWS KMS key are blocked or disabled, the Secret Server website becomes inaccessible until the prior credentials are restored by the customer.

Introduction

Secret Server protects your secrets using a master encryption key, as well as an additional intermediate encryption key that is unique for each secret. These effectively act as internal passwords that Secret Server itself needs to unlock your data, for example any time you view or update a secret.

Key Management in Secret Server Cloud (SSC) allows you to add an additional layer of encryption using a third-party provider to protect these encryption keys for added protection and control. To do this, you must first set up your own encryption key with a third party that you fully control, and then provide Secret Server limited access to it. This external encryption key is used to protect the Secret Server encryption keys. You can revoke Secret Server's access at any time if the need arises, rendering Secrets unusable.

> Once enabled, beware that if you delete your external third-party encryption key, or the credentials you gave Secret Server no longer work. *You will not be able to access your existing Secrets, and even Delinea will not be able to help!*

You can change your key management configuration through Secret Server's Web interface or by using the REST API. If key management has already been enabled, you can switch to a new configuration or disable key management completely. To make any change, your existing key management configuration **must still be valid**, so your secrets and the master encryption keys can be converted to the new configuration. Your new settings are validated before they can be saved.

Secret Server Cloud currently supports Amazon's Key Management Service.

Amazon Key Management Service

Key Management Service (KMS) is a managed service provided by AWS that allows you to create, manage and use encryption keys for your applications and services. With KMS, you can create symmetric keys or asymmetric keys to encrypt and decrypt data. These keys can be used to protect sensitive data such as passwords, credit card numbers, or personally identifiable information (PII).

A KMS (Key Management Service) key is a cryptographic key used to encrypt and decrypt data stored in AWS (Amazon Web Services) services such as S3, EBS, or RDS. KMS keys are stored securely in the AWS Cloud, and you can control access to them by using IAM (Identity and Access Management) policies. You can also use KMS to audit key usage and generate key usage reports.

Configuring Key Management

To enable key management, you will first create an encryption key with your third-party provider, then an API account that Secret Server will use in order to access the key. After the external encryption key is setup, you will update Secret Server with the details.

> Changing your key management settings will trigger "maintenance mode" and a secret key rotation that will re-encrypt all your secret keys. No one will be able to access secrets until the rotation finishes, and it must finish successfully before further key management changes can be made.

Navigate to Secret Server's key management page by clicking **Admin > All > Key Management**.

Here you can change your key management settings, as well as view the audit history showing all key management updates.

Key Management Providers

Secret Server Cloud currently supports one provider, AWS Key Management Service. More providers may be added over time. Azure's KeyVault service is not a viable provider at this time due to slow speed limits when using strong encryption keys (such as 4096-bit RSA with HSM).

AWS Key Management Services Pricing

Please see AWS Key Management Service Pricing.

Secret Server Cloud requires one AWS Key ("CMK"), and the number of requests per month will vary depending on how often secrets are accessed.

Procedure

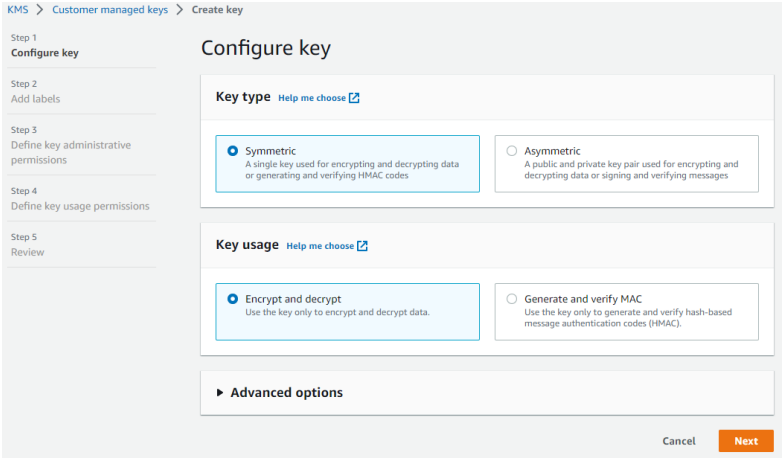
> Changing your key management settings triggers Secret Server Cloud maintenance mode and a secret key rotation that re-encrypts (or decrypts) all your secret keys! No one can access secrets until the rotation finishes, and it must finish successfully before further key management changes can be made.

Task 1: Setting up the Encryption Key and IAM User in AWS

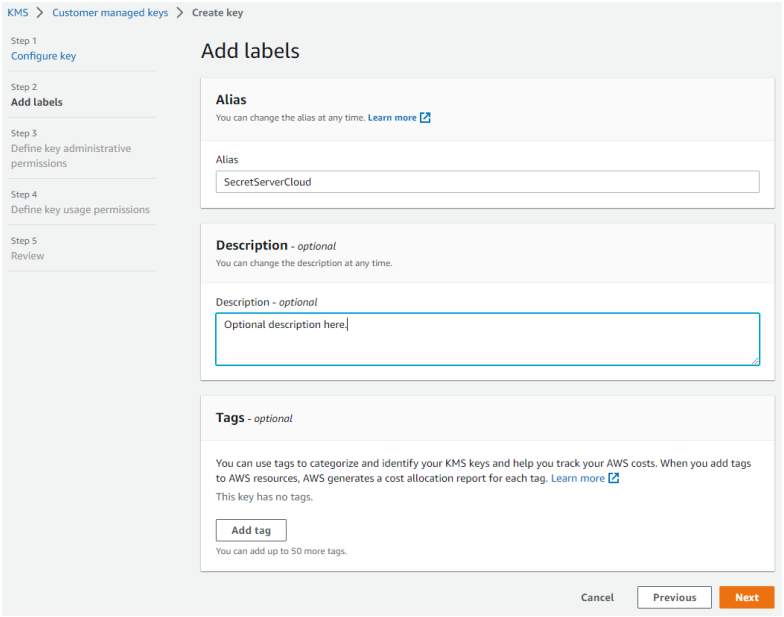
1. Log into the AWS Console website at https://console.aws.amazon.com/.
2. Under **Services**, search for **IAM** (Identity and Access Management). This is where you will configure both your encryption key and an IAM user for Secret Server to use to access the encryption key.
3. Click the **Users** button on the left menu.
4. Click **Add User** button.
5. Type a name (such as *SecretServerCloud*) in the **User Name** text box.
6. Click to select the **Programmatic Access** check box in the **Access Type** section.
7. Click the **Next: Permissions** button (on the Permissions page, no special permissions are needed). The Permissions page appears.

8. Click the **Next: Tags** button. The Tags page appears.
9. Click the **Next: Review** button (on the Permissions page, no special permissions are needed). The Review page appears.
10. Click the **Create User** button. A Success page appears confirming the user was created. Both the access key ID and the secret access key appear (click the **Show** link).
11. Click the **Download .csv** button to save the credentials

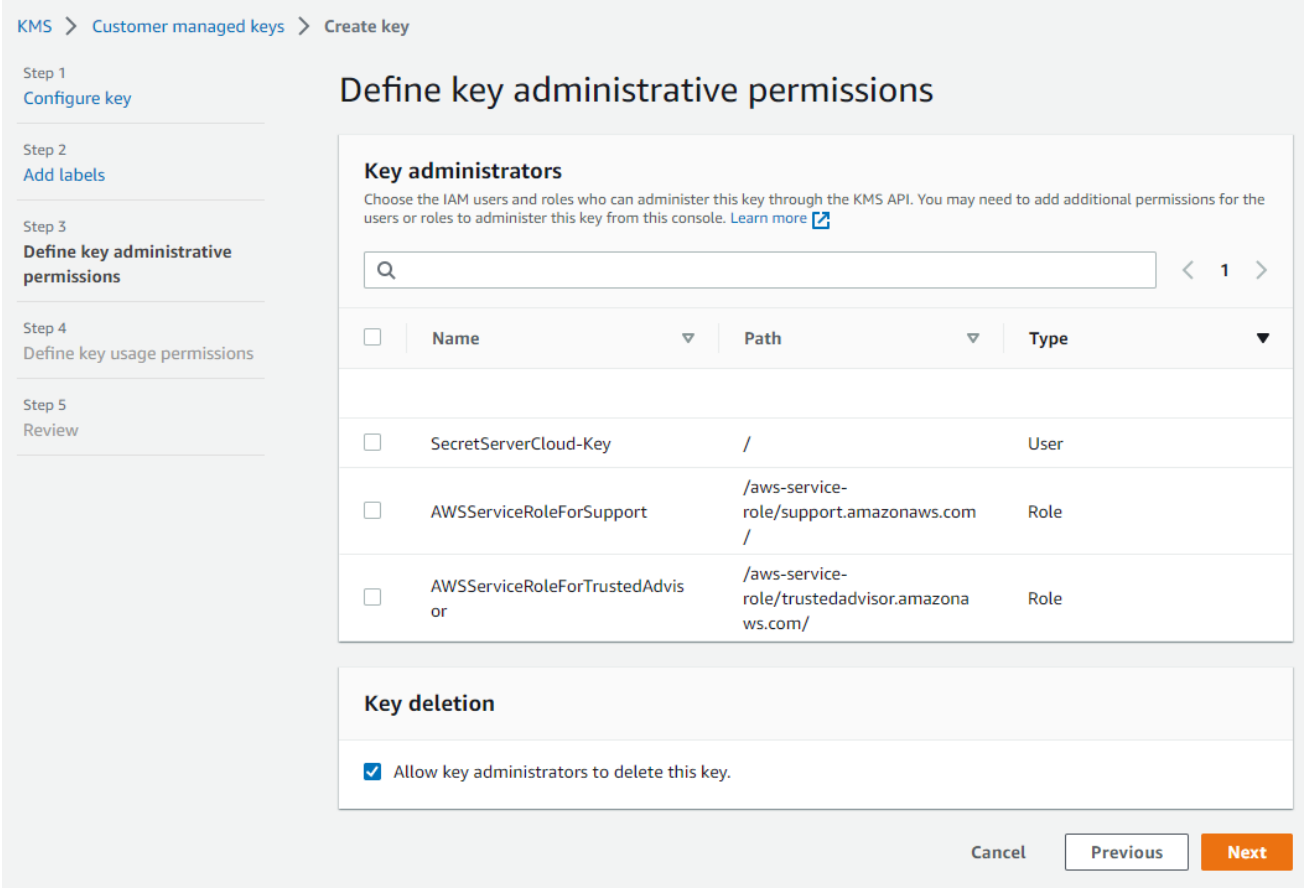
    Important: Be sure to **save both the access key ID and the secret access key**! If you lose them, you can never view the secret access key again. Even after you enter them in Secret Server Cloud, you cannot retrieve the secret access key.
12. Once the download completes, click the **Close** button.
13. Under **Services**, search for **Key Management Service**.
14. Click the **Customer managed keys** link in the left menu.
15. Click the **Create Key** button. The Configure Key page, the first page of the Create Key wizard, appears:



16. Ensure the **Key type** selection button is set to **Symmetric**.
17. Ensure the **Key usage** selection button is set to **Encrypt and decrypt**.
18. Click the **Next** button. The Add Labels page appears:

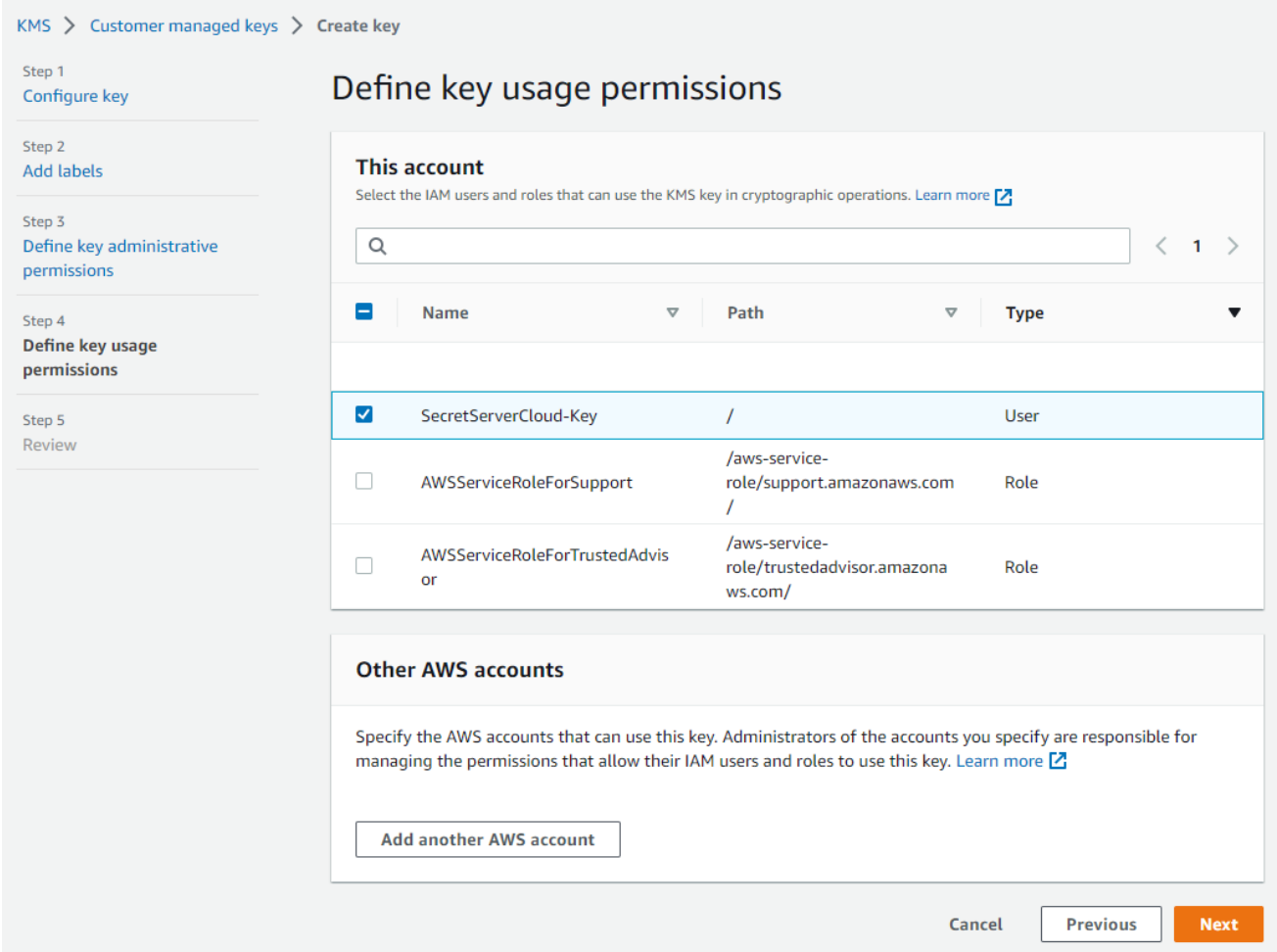


19. Type `SecretServerCloud` in the **Alias** text box.
20. (Optional) Type a description in the **Description** text box.
21. (Optional) Click the **Add** tab button to add KMS tags. Click the **Learn More** link for more about tags.
22. Click the **Next** button. The Define Key Administrative Permissions page appears:



Leave the page as is.

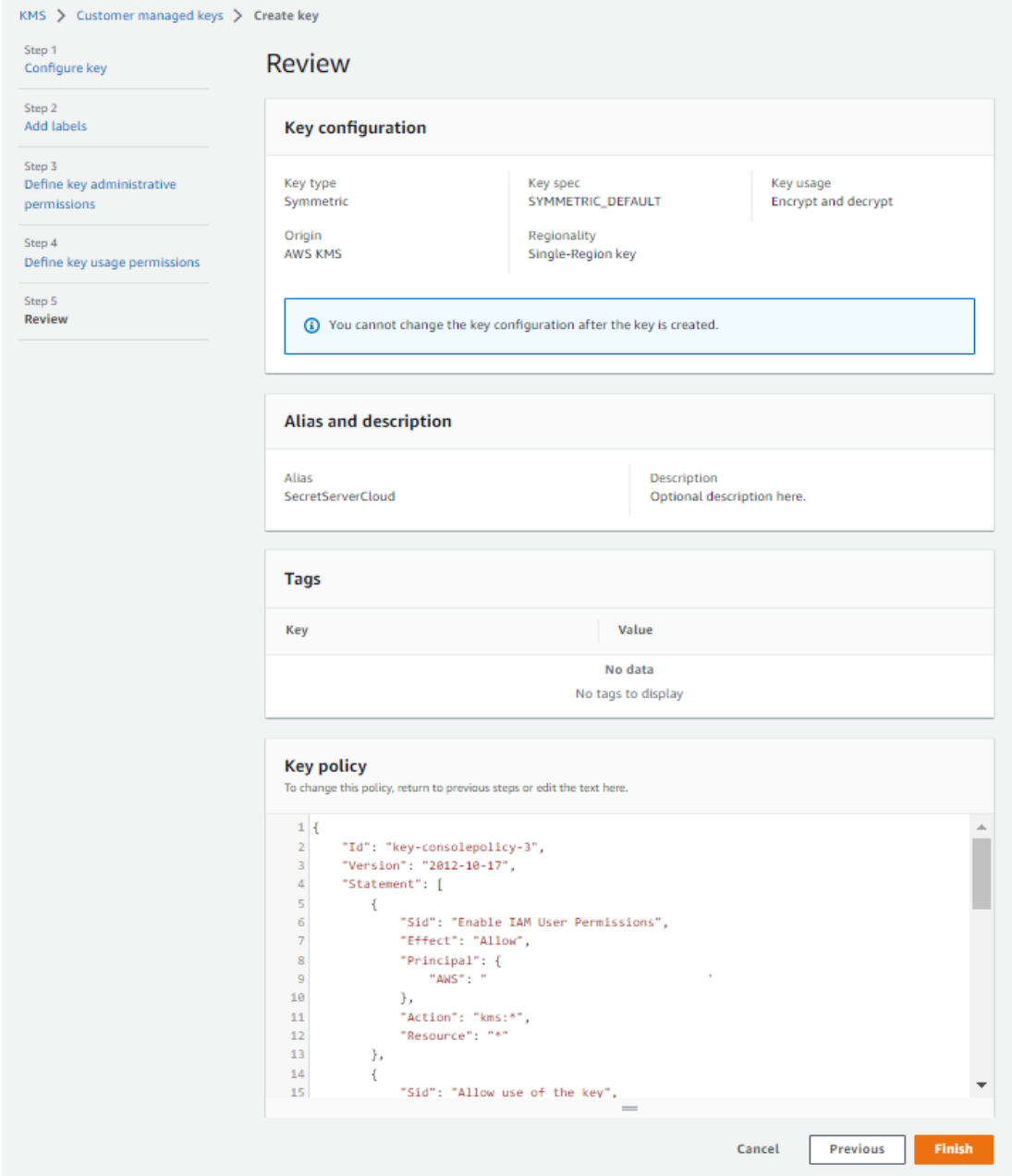
23. Click the **Next** button. The Define Key Usage Permissions page appears:



24. Click to select the check box next to the SecretServerCloud-Key name in the table to give that user access to the key.

    Important: Do **not** give access to the user you created earlier for Secret Server Cloud. It is unnecessary for Secret Server to have administrative access to the key.

25. Click the **Next** button. The Review page appears:



26. Ensure the settings are as desired.
27. Click the **Finish** button. The new key appears in your Encryption Keys list.
28. Click to select the new key in the list. The Summary section on the key's page appears.
29. Copy and save the contents of the read-only **ARN** text box. You will need it later.

> AWS supports automatically rotating this key every year. You can change that setting on this page in the Key Rotation section (select the "Rotate this Key every year" check box). Once rotated, the key management settings in Secret Server will not require further changes, and your existing secrets can still be accessed by the old encryption settings. However, only new secrets will be created under the new version of the encryption key, and you must perform a secret key rotation inside Secret Server Cloud if you want to update all secrets to use the new version of the AWS key.

> As a security best practice, we recommend performing a secret key rotation inside of Secret Server Cloud on a regular basis to refresh the encryption keys on your Secrets. Go to Admin > Configuration > Security, and click Rotate Secret Keys.

Task 2: Adding Encryption Key and User Details in Secret Server

1. In Secret Server Cloud, go to **Administration > Key Management**. The Key Management page appears.
2. Click the **Edit** button. The page becomes editable.
3. Click the **Key Management Type** dropdown list to select **Amazon KMS**.
4. Type your AWS key details that you saved earlier in the remaining four text boxes.
5. Click the **Save** button.

Task 3: Secret Key Rotation

1. Once you save your changes, your new settings are validated and a secret key rotation is triggered.
2. View the progress of the rotation:
   a. Go to **Admin > Configuration**.
   b. Click the **Security** tab.
   c. Go to the **Key Rotation** section.
3. Later you can repeat the process to change the AWS encryption key, or you can select **None** for the **Key Management Type** to disable it completely.

Secret Server Key Management via the REST API

Secret Server Cloud has a REST API for retrieving or updating your key management configuration. For details:

1. Log on your Secret Server Cloud instance.
2. Click the question mark icon in the top right corner and select **Secret Server REST API Guide**.
3. Click on the **Documentation for REST API** document link for your authentication style, normal tokens or Windows Integrated Authentication.
4. Search for `KeyManagement` to view that section of our API.

Important: When changed via the API, maintenance mode and a secret key rotation still occur.

Secret Server Cloud Text Field Character Limits

These values are valid for Secret Server Cloud only. Platform groups, roles, and users have different limits.

Character limits as of June 2024:

- Folder Name: 128 characters. Error message: "The folder name must be 128 characters or less."
- Group Name: 250 characters. Error message: "Application Error."
- Local User Display Name: 256 characters.
- Local User password: 500 characters.
- Local User Username: 128 characters.
- Request field: 600 characters. This is a comment field used when requesting access to a secret. No error message is given when surpassed. Instead, it prevents exceeding 600 characters. If you paste in more than that, it clips the text and still lets you save.
- Role Name: 255 characters. Error message: "Please choose a Role name with between 1 and 255 characters."
- Secret Name: 1,992 characters. Error message: "Invalid Secret Name."
- Secret Note Field: 9,999 characters. Error message: "Secret item value exceeds max length characters."
- Secret Text Fields: 9,999 characters. Error message: "Secret item value exceeds max length characters."

Secret Server Cloud Offboarding

This topic only applies to standalone **Secret Server Cloud**.

Privacy Policy

See the Delinea Privacy Policy for details on how your data is safeguarded.

Data Protection

At all times, even after your license expires, Delinea maintains strict security controls of your data. Only a tiny group of operations staff can access it and only with approval from at least one other person. You can get a SOC2 audit report for your instance, which requires an NDA, by emailing the RFP Helpdesk. The NDA is required because proprietary Delinea information is present in the report.

Your Data When Your Subscription Ends

Once your subscription ends:

1. Your instance enters a two-week grace period.
2. After the grace period expires, the instance is scheduled for deletion two weeks in the future. Your instance is inaccessible but all data is retained.

3. Once the deletion date arrives, your instance is eligible for deletion and can be deleted at any time at Delinea's discretion. In most cases, Delinea retains customer data substantially beyond the deletion date.

If desired, you can request that Delinea deletes your instance right away, and we will promptly respond. **Once deleted, your instance is not recoverable and your data is gone.**

Session Recording Overview

Delinea offers basic or advanced recording options, enabled by different tools and configurations that capture varying levels of content.

Basic session recording supports logging keystroke metadata for RDP and SSH sessions without requiring an agent across both Windows and Mac environments. Users can search for keystrokes, and the session playback interface displays this additional activity information.

Advanced session recording offers more granular capabilities and process metadata.

The following tools record videos:

- Protocol Handler
- Protocol Handler on a session connector server
- Web Password Filler
- ASRA (Advanced Session Recording Agent)
- Privileged Remote Access

The following tools record keystrokes:

- RDP Proxy
- SSH Proxy
- ASRA (Advanced Session Recording Agent)
- Protocol Handler on a session connector server

The following tools can record process metadata:

- ASRA (Advanced Session Recording Agent)

Basic Session Recording

macOS Catalina requires additional configuration to use basic session recording. See "macOS Catalina Security" on page 1233.

Basic session recording is a licensed feature in Secret Server. It relies on the protocol handler configured on client machines through Secret Server's launcher. Using the launcher, Secret Server captures second-by-second screenshots on the client machine during a user's recorded session. These images of the user's screen are compiled into a video that can be downloaded and played back for auditing and security purposes. Activity recorded

in the session is based on screen changes only. This is valid for Mac Launchers. RDP/SSH launchers upload recorded video segments every second. See Session Recording Example Architectures for more details.

The Secret Server Session Connector uses a modified Protocol Handler that supports keystroke recording.

Session monitoring allows administrators with the Session Monitoring permission to view all active launched sessions within Secret Server. If session recording is enabled on the secret, an administrator can watch the user's session in real time.

Admins can search through active and ended sessions. To review and search through sessions go to **Admin > Session Monitoring**.

Searching across sessions can search the following data. To select what data is searched across check the options on the search filters on the left-hand side.

Some search filters require additional components to be installed or configured:

- **Proxy Session Client Data**: Search within keystroke data of proxied SSH sessions. Requires that the SSH proxy is enabled and SSH sessions are using it.
- **RDP Keystroke Data**: Requires the ASRA be installed on the target or the RDP proxy or session connector components are available.
- **RDP Application Name**: Requires the additional ASRA be installed on the target.

To view a recording, click the camera icon on the session. The Watch Session Recording page appears.

If there is logged session activity, such as keystroke or application data, then you can search through session activity and jump to points within the video playback. The playback also displays an activity map to show points of high activity, such as screen changes, keystrokes, and processes started and stopped.

Selecting an activity in the grid also shows additional details below such as the full folder path where the application started and the user that performed the operation.

> SSH Keystroke data is shown in one-minute segments. In a short session of less than minute, the "jump to" only goes to the beginning of the video.

For active sessions, there are two actions that can be taken:

- **Watch Live:** When session recording is turned on for the secret and admin can view and replay the user's activity.
- **Terminate:** Sends a message to the business user or terminates their session. The business user sees an alert dialog pop up on their machine with the message. Session Recording does not need to be enabled for this to work. However, when Session Recording is enabled, the functionality exists on both the secret itself, and the session in Session Monitoring. For ended sessions admins can watch the recorded video and view the SSH log if session recording was turned on for the secret.

Advanced Session Recording

Advanced Session Recording (ASR) is a licensed feature of Secret Server that adds capabilities to those offered by basic session recording. You install the Advanced Session Recording Agent (ASRA), which uses the Remote Desktop Protocol, on any client machine where you want more information from the sessions recorded.

ASR is not available to those using our Mac launcher.

Older ASRAs (earlier than 7.7) only work if a distributed engine configuration is enabled with RabbitMQ or MemoryMQ installed.

ASR enhances the launcher sessions, which typically only include screenshots, keystrokes, and process activity. ASR features include:

- Screen Capture: The Secret Server launcher records second-by-second screen images compiled into a playback video of the user's session. This is essentially the same as basic session recording.
- Logged Processes: The ASRA logs all processes started and stopped during a user's session.
- Recorded Key Strokes: The ASRA records all user keystrokes during the session, which can be disabled.

In addition to those, ASR includes these enhanced video playback features:

- Searchable Video: You can search video activity to find locations where specific activities, such as specific keystrokes or ran processes.
- Enhanced Playback: Sessions recorded using ASR display additional data on playback, such as the current active window, the used processes, and keystrokes in the session.
- On-demand video processing
- Recording all sessions
- Inactivity timeout
- Maximum session-length protection

The Windows protocol handler encodes your session in WEBM format in real time and sends the recording to Secret Server. There is now an "Enable On-Demand Video Processing" option in Secret Server which leaves the recordings in WEBM format, which Chrome and Firefox can playback without any further processing, saving server processing time. If an on-demand recording is viewed with Internet Explorer or Edge (which do not support WEBM playback), you can click a "Request Video Processing" button and the video will be converted to H.264/MP4, which they can then play. If "Enable On-Demand Video Processing" is not checked, then all sessions recorded by the Windows protocol handler will be automatically converted to H.264/MP4.

The Mac protocol handler does not yet support this feature, so any recordings created with it are converted to the chosen legacy video codec format. We recommend H.264/MP4. You can set the advanced session recording agent to "Record All Sessions." If someone logs into a server directly without launching from Secret Server, or even logs in at the console, the full session is recorded, including metadata.

See "Installing the Advanced Session-Recording Agent" on page 1240.

Session Recording Tab

The Session Recording tab contains the following configuration options:

- **Enable Deleting:** After the "Days Until Deleting" value, Secret Server deletes the videos from disk. Secret Server
- **Enable Moving to Disk:** After the "Days Until Moved to Disk" value, Secret Server can move videos from the database to an archive path on disk.
- **Enable Session Recording:** Enable session recording for launched sessions.
- **Save Videos To:** By default, videos are stored in the database, Secret Server can also store them directly to a network share. This network share must be accessible from all Web servers that Secret Server is installed on. Secret Server
- **Video Code:** Specify the codec to use to create the videos from the launcher screenshots. This codec must be installed on the Web server (or servers if clustering is enabled) that Secret Server is installed on.
- **Retry Failed Videos:** A failed video is any recording that did not capture the complete session. The system will try and reprocess any video in the database that has a "failed" status but, it's basically just going to do the exact same thing that it tried previously. It's not really a guarantee that all videos that are retried will actually be successful. To check for failed videos, navigate to **Session Recording Errors** in the Reports section or relevant logs that would show where videos may have failed.

> The Microsoft Video 1 codec is for testing only and does not support in browser playback. Sessions encoded with Microsoft Video 1 can still be downloaded for review.

For details on the settings in the Login and "Local User Passwords" tab, see "Configuring Users" on page 1273.

Advanced Session Recording Requirements

> This applies to ASRA and Secret Server. See below for additional details.

> All cores are physical unless otherwise noted.

Table: Advanced Session Recording Requirements

Web Server (Secret Server)	Database Server (SQL Server)	ASRA (Client Machines)
8 CPU Cores	8 CPU Cores	2 CPU Cores
32 GB RAM	32 GB RAM	16 GB RAM (4 GB for the agent itself)
50 GB Disk Space	100+ GB Disk Space	25 GB Disk Space
Windows Server 2012 or 2016	Windows Server 2012 or 2016	Windows 7 or newer

Web Server (Secret Server)	Database Server (SQL Server)	ASRA (Client Machines)
IIS 7 or newer	SQL Server 2012 or newer	
.NET 4.6.1 or newer		

Basic Session Recording Requirements

All cores are physical unless otherwise noted.

Table: Basic Session Recording Requirements

Web Server (Secret Server)	Database Server (SQL Server)
8 CPU Cores	8 CPU Cores
16 GB RAM	16 GB RAM
25 GB Disk Space	100+ GB Disk Space
Windows Server 2012-2022	Windows Server 2012-2022
IIS 7 or newer	SQL Server 2012 or newer
	.NET 4.6.1 or newer

Windows Server 2022 supports RDS CALs licensing.

Caveats and Recommendations

General

System requirements apply to both physical and virtual machines.

- Delinea does not support these Web servers:
- Any Client OS
- Domain Controllers
- SharePoint Servers
- Small Business Server (SBS)
- Windows Server Essentials
- For best performance, we recommend using dedicated (clean) servers for hosting Delinea products.

- If .NET and IIS features are not already installed on the Web server, the Delinea Installer adds and configure them automatically.

Components Supporting Session Recording

Things that can record video:

- Protocol Handler (PH)
- PH on a session connector server
- Web Password Filler (WPF)
- Advanced Session Recording Agent (ASRA)
- Remote Access Service (RAS)

Things that can record keystrokes:

- Remote Desktop Protocol (RDP) Proxy
- Secure SHell (SSH) Proxy
- ASRA
- PH on a session connector server
- RAS

Things that can record process metadata:

- ASRA

Things that can record nothing:

- Secret Server Session Connector (SSSC)

Database

- Database disk storage depends directly on how many recorded videos are stored to disk. For active users, we recommend you **use a 1 TB shared or local drive for archival or storage space**. For light users, we recommend beginning with 300 GB. Monitor your disk space usage closely, and tailor it for best results.
- **Carefully consider how quickly your allotted storage might be exhausted**. Once again, it is highly variable, but you might expect around 15 hours of recording per GB of storage. Using the example of encoding capacity used in the Session Recording section, if you wanted to record one year of usage by your 60 8-hour users, you would need around 11 TBs of storage (given vacations and holidays). Our recommended 1 TB would last nearly a month in that scenario. A session retention policy using the automatic deletion feature is likely your best option.
- If MS SQL Server is not already installed on your database server, the Delinea Installer can setup SQL Express on the Web server; however, **SQL Express is only for trials and sandbox environments**. Delinea does not support using SQL Express in a production environment due to size and performance limitations.

Please see Microsoft documentation on SQL Express at: https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-2017

Network Bandwidth and Video

- For Secret Server 10.6 ASR requires around 300 Kbps. Older versions of Session Recording require 1-3 Mbps.

> Our Mac launcher uses the older bit rate.

- Session recording bandwidth requirements vary widely based on monitor resolution and image complexity--higher resolutions and more complex images (simpler screen images compress better) use more bandwidth. For example, with a 1024×768 screen resolution, the required network bandwidth is typically between 0.1 Mbps and 1 Mbps.
- If your connection cannot support the needed bandwidth, the session data is still transmitted, but it takes longer to process each session.
- If a user tries to cancel the transmission, this activity appears in the audit record for the Session Recording Secret.
- All sessions are recorded at 1080p.

> Before Secret Server 10.6, session recordings 1080p or higher were not supported due to a limitation in Microsoft IIS. The session video would be recorded but may have been corrupted.

- Sessions are recorded using the H.264 MPEG-4 codec.

Session Recording

- Server hosting session recording requires fixed RAM and disk space. We strongly recommend that you **do not apply dynamic settings**.
- **Do not record more sessions than you can encode**. If more concurrent sessions are recorded than the system can process, the sessions wait in a queue and are processed when enough server resources become available, which could be in a very long time or perhaps never if your storage is overwhelmed.
- The frame rate we can encode varies dramatically based on many factors, so **testing what encoding rate your session recording configuration can sustain is a must**. From there, you can get an idea of what is possible. For example, let us say you found that we can process 20 FPS on average on your Xeon processors. Given that rate, we could encode around 1 minute of a session recording in 3 seconds, or 1 hour in 3 minutes, or 1 day in 72 minutes--giving you perhaps 480 session hours per day. You could then parse that figure based on your typical usage to arrive at a maximum potential usage, for example, 60 people doing 8-hours of session recording.
- Typically, you can record **up to one hundred sessions at a time per web node**, load balanced, which should handle large use cases.
- CPU usage during video processing varies depending on concurrent users and recording length. We recommend that you **closely monitor CPU percentages on your web server** during video processing, as well on your client machines during recording, to increase CPU count for machines, if needed.
- We recommend that you **set up RabbitMQ as the backbone service bus** in session recording environments. To setup RabbitMQ, see "Installing RabbitMQ" on page 93.

For the highest scalability and reliability, Delinea recommends using RabbitMQ. MemoryMQ is an easier but less capable alternative and can be used for trials and proof of concepts but **should not** be used for production environments. Two exceptions are very small deployments and customers that do not use open-source software for compliance reasons.

Session Recording Web Node Connectivity Failures

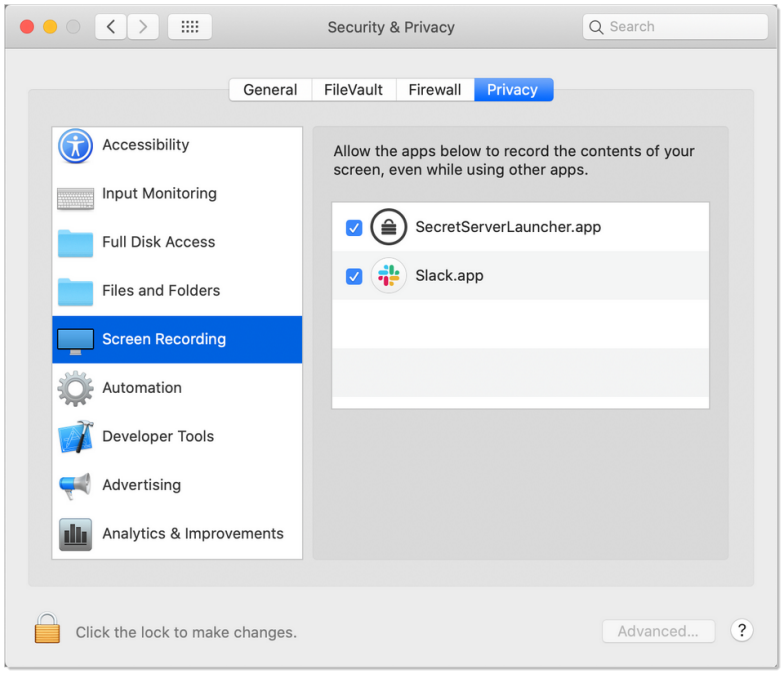
In the event that a session recording web node is disconnected, there is a caching mechanisms:

- Protocol Handler - The information is cached on the machine when just using basic Session Recording.
- Web Password Filler - When recording a browser session if the Web Password Filler loses connectivity to Secret Server's web server over 443, the session is disconnected, and whatever had been recorded up to that point is sent to Secret Server when the connection is restored.
- Advanced Session Recording Agent - If the connection is lost over port 443, the session is unaffected. ASRA will record locally and then when connectivity is restored it will send the data to Secret Server.
- Session Recording with the Proxy - You can block port 443 without any issues when using the proxy but, if you were to block the Proxying port the session would be disconnected.

macOS Catalina Security

macOS Catalina enforces security policy around screen recording. To use the session recording feature of the Delinea launcher on MacOS Catalina, you must first:

1. Go to **System Preferences > Security & Privacy > Screen Recording** on your Mac.
2. Allow recording for the SecretServerLauncher.app:



Configuring the Maximum Concurrent Recording Sessions per Web Node

To set the maximum number of concurrent recording sessions allowed per web node, follow the procedure below on your Secret Server web server node dedicated to session recording:

1. Navigate to the `web-appSettings_config` file (default location `C:\inetpub\wwwroot\SecretServer\web-appSettings.config`).
2. Right-click the file to open it with Notepad.
3. Before the final `appSettings` line, insert the following string:

   ```
   <add key="PrefetchCount.ConvertVideoMessage" value="7"/>
   ```
4. At the end of the string, set the value for the maximum number of concurrent sessions you want for this node. In the example above, the maximum number of sessions is set to `7`.
5. Save the notepad file.
6. Restart IIS on the server.

Configuring Session Recording

Overview

Session recording allows you to record an RDP or PuTTY session, with optional metadata, and play it back in Secret Server.

The Windows protocol handler encodes your session in WebM format in real time and sends the recording to Secret Server. There is an "Enable On-Demand Video Processing" option in Secret Server which leaves the recordings in WebM format, which Chrome and Firefox can playback without any further processing, saving server processing time. If an on-demand recording is viewed with Internet Explorer or Edge (which do not support WebM playback), you can click the "Request Video Processing" button and the video is converted to H.264/MP4, which they can then play. If "Enable On-Demand Video Processing" is not checked, then all sessions recorded by the Windows protocol handler are automatically converted to H.264/MP4.

The Mac protocol handler does not yet support this feature, so any recordings created with it are converted to the chosen legacy video codec format. We recommend H.264/MP4.

You can set the advanced session recording agent to "Record All Sessions." If someone logs into a server directly without launching from Secret Server, or even logs in at the console, the full session is recorded, including metadata.

Configuration

1. Go to **Admin > Configuration > Session Recording**.
2. On the **Session Recording** tab, click the **Edit** button.
3. Ensure the **Enable Session Recording** check box is selected.

For testing and proof of concept deployments, Secret Server's "Internal Site Connector" on page 786 is sufficient for session recording. For production deployments we strongly recommend "Installing RabbitMQ" on page 93 for a more-robust message queue.

Using Legacy Video Codecs

You can select a legacy video, but it will only apply to sessions recorded by the Mac protocol handler. Delinea recommends the H.264 codec, which was available starting in Secret Server 10.5.000003 because it produces the highest quality videos and requires no additional installation. If you want a different legacy codec, ensure that the codec you select is correctly installed on the same machine as Secret Server. It does not need installation on any client machines, where the session recording is occurring.

Available legacy codecs:

On Windows Server 2008 and above, you can install Window Media Player by adding "Desktop Experience" from the features of Server Manager.

- Microsoft Video 1 (testing only): Microsoft Video 1 is deprecated in favor of Microsoft Video 9 and should not be used for production. Microsoft Video 1 does not support browser-based playback of sessions.
- Microsoft Video 9: High compression level and quality. Requires Windows Media Player. This option produces comparable video sizes to Xvid for moderate activity in an RDP session.
- VP8: High compression level and quality. VP8 is bundled with Secret Server. This option produces comparable sized video to Xvid for moderate activity in an RDP session.
- Xvid: Provides similar quality and compression to DivX and is freely available. This option produces approximately 20 MBs of video for 1 hour of moderate activity in an RDP session. See https://www.xvid.com/

Enabling Session Recording on Secrets

You must enable session recording on the Security tab for each secret. Once session recording is enabled, Secret Server records that session when the launcher is used.

To view the recorded session after it is completed, click the **View Audit** button on the secret screen and then the **View Session Recording** link in the **Details** column.

You can also search recordings from the Session Monitoring page under **Admin > Session Monitoring**.

The Session Monitoring page lets users search and filter sessions based on session data, secrets, users, groups, launcher type, date, and folders. This page is also where any recordings appear when using the Record All Sessions option (see below), because such recordings are not tied to a specific secret.

Browser playback is only supported in Secret Server 10.2 and higher. Older versions of Secret Server prompt the user to download the recording.

To view a session, click the camera icon to the right of it. This takes you to the Web playback interface. The video playback shows an activity map to quickly skip to sections of higher usage.

As noted above, if using the "On-Demand Video Processing" option, Chrome and Firefox can play the video. If you try to view an on-demand video using Internet Explorer or Edge, a warning message appears.

If you click the **Request Video Processing** button, the recording is converted from WebM to H.264 as soon as possible, allowing IE/Edge to play it back.

Extending Session Recording with Custom Launchers

You can configure Secret Server with custom launchers to run arbitrary programs, which can then be recorded by session recording. To do so:

1. Define a custom launcher:
    a. Go to **Admin > Secret Templates > Launchers**. The Manage Launcher Types page appears.
    b. Click the **Create** button.
    c. Leave the **Launcher Type** dropdown list set to **Process**.
    d. Type a name for the custom launcher in the **Launcher Name** text box.
    e. Type a process name in the Process Name text box.
    f. (optional) Type process arguments in the Process Arguments text box.
    g. Customize other Options as needed.
    h. Click the **Save** button.
2. Associate the launcher with a secret template:
    a. Go to **Admin > Secret Templates**. The Manage Secret Templates page appears.
    b. Click the template dropdown list and select the desired template.
    c. Click the **Edit** button.
    d. Click the **Configure Launcher** button. The Secret Template Edit Launcher Configuration page appears.
    e. Click the **Add New Launcher** button.
    f. In the **Launcher Type to use** dropdown list, select your custom launcher.
    g. Customize any other options as needed.

Secret Server 10.8 added two new options to custom launchers:

Record Multiple Windows Option

If this option is not checked, only the main window of the main launcher process will be recorded (this was always the behavior prior to Secret Server 10.8). If it is checked, multiple windows as well as child processes are recorded.

Without this enabled, the main window of the main process sometimes does not show anything useful, depending on the application, resulting in a blank recording. With this enabled, recordings are generally more accurate. This also applies to applications that can open or undock separate windows or those that launch additional processes, such as an application launching PowerShell and then launching other applications from the command prompt.

Record Additional Processes Option

Here you can type an optional comma-separated list of processes to record if found, running under your same user account, that are not started or terminated by the custom launcher. "Track Multiple Windows" must be enabled for this option to be available.

In the example above of launching PowerShell and then opening Notepad, if "Track Multiple Windows" is enabled, both PowerShell and Notepad would be recorded automatically, because the OS can tell that Notepad is a child process of PowerShell. This even works multiple levels deep–for example, launching PowerShell, then the command prompt, and then launching in PowerShell again, finally followed by Notepad.

In some cases, though, you may wish to record an additional process that was already running before the custom launcher was launched or may want to start running one later. To this end, any process names specified in this option are checked for periodically, and recording is attempted on them as well.

Example

If you wanted to run an X11 server such as Xming and then PuTTY with X11 forwarding, you could configure a custom launcher with these values:

Process Name: `C:\Program Files\PuTTY\putty.exe` Process Arguments: `-X -ssh $MACHINE -l $USERNAME -pw $PASSWORD` Record Additional Processes: `Xming.exe`

In this case, Xming should already be running before the launcher was used and would remain running after the session has ended. It would have no parent/child relationship with PuTTY at all. However, while the launcher session was active, any windows it spawns would still be recorded, allowing the X11-forwarded applications to be recorded, not only the PuTTY window.

Advanced Session Recording

Metadata Recording

By default, session recording creates videos of the launched session. Secret Server supports logging additional metadata, such as keystrokes for RDP and SSH sessions. When these options are enabled, users can search for keystrokes or applications across sessions, and the session playback interface shows additional activity information.

Remote Desktop session metadata requires Secret Server and the advanced session recording feature, which in turn requires an installation of an advanced session recording agent (ASRA) on the target servers. See "Installing the Advanced Session-Recording Agent" on page 1240.

SSH keystroke data relies on the Secret Server SSH Proxy. This can be enabled under **Admin >SSH Proxy**. See "SSH Proxy Configuration" on page 831 for more information. Once proxying is enabled recorded SSH sessions will log SSH traffic which can be searched and is displayed in the session playback interface.

RDP Proxy and Advanced Session Recording Agent (ASRA) are able to capture RDP metadata.

Record All Sessions

You can configure the ASRA to record all sessions. This causes it to record video, keystrokes, and process metadata for anyone logging into the server, even when not using Secret Server, including logging into the console. Since these recordings are not tied to any specific secret, you must go to the **Admin>Session Monitoring** page to view them.

Session Recording Settings

Under **Admin >Configuration>Session Recording** there are several settings for configuring how Secret Server handles session recordings:

Hide Recording Indicator

When viewing a secret, the launcher icon normally indicates if the session will be recorded or not via the recording icon. When launched into, a notification window also informs the user that their session is being recorded. If "Hide Recording Indicator" is checked, users cannot tell which secrets have recording enabled based on the icons, and if they launch a recorded session, they will not be warned that their session is being recorded.

Enable On-Demand Video Processing

The Windows protocol handler encodes the recording on the fly in WebM format and streams the video to Secret Server. Once the session has ended, Secret Server reconstructs the video and leaves it in WebM format, which Chrome and Firefox can natively play back.

> WebM is an audiovisual media file format that is a royalty-free alternative to HTML5 audio and video.

Internet Explorer and Edge currently have issues playing back WebM videos, so if you are using those browsers and try to view an on-demand recording, you are presented with a "Request Video Processing" button, which converts the video to H.264/MP4, as soon as possible, which IE or Edge can then play back.

If this option is not checked, all sessions recorded by the Windows protocol handler are converted to H.264/MP4 automatically. If you have many IE or Edge users, Delinea recommends leaving this option unchecked, but this will increase the processing time of videos and increase the load on your Secret Server servers that have the Session Recording role enabled.

This setting has no effect on sessions recorded with the Mac protocol handler, which is always encoded using your legacy video codec choice.

Enable Inactivity Timeout (Minutes)

If enabled, if a session appears idle, users are given a five-minute warning that they will be disconnected. A prompt appears that lets them choose to disconnect immediately or to continue the session. If no response is received, the session is disconnected five minutes later.

> This feature was added in Secret Server 10.6.26 and is currently only supported in the Windows protocol handler (not Mac).

Max Session Length (Hours)

This sets a hard limit to how long a recorded session may last. This includes both launched from Secret Server, as well as recorded sessions if using ASRA and the "Record All Sessions" option. This option helps prevent accidental recordings over the weekend, or even longer, if someone forgets to disconnect their session.

This feature was added in Secret Server 10.6.26 and is supported by both the Windows and Mac protocol handlers.

Use Hardware Acceleration

If enabled, when processing H.264/MP4 files, this setting makes Secret Server attempt to use hardware acceleration for video processing if possible (GPU or CPU). Delinea recommends this setting is always enabled because Secret Server will fall back to not using hardware acceleration if necessary.

This feature was added in Secret Server 10.6.0.

Save Videos to

This configuration includes:

- **Database**: Stores the information from a recorded session as encrypted data to your database.
- **Disk**: Stores the recorded session as a video file directly to the specified folder path.

Archive Location Dependent on Site

If you save recordings to disk, enabling this option lets you pick a separate path for each of your sites. This is useful in large environments that need many recordings spread out across multiple devices and locations.

See below for a note about using network shares for storage.

Folder Path

If you save recordings to disk, this is where they are saved. If you use the "Archive Location Dependent on Site" option, this is the default storage location for newly added sites, until you customize their folder path to something else.

See below for a note about using network shares for storage.

Encrypt Archive on Disk

This setting encrypts the session videos when stored on disk. Videos stored on disk are played back through the Secret Server UI but cannot be viewed directly from the file system.

Enable Archiving to Disk

After the specified number of days have passed, all recorded session information in your database is transferred to the specified folder path as video files and cleared from the database.

Session Recording Retention Schedule

Enable Deleting

After the specified number of days have passed, all recorded videos in your database will be cleared and video files in your archive path will be deleted.

Setting Notes

- To use "Save Videos to Disk" or "Archive to Disk," the Application Pool service account must have write permission to the specified file path.
- To delete videos from the archive path, the Application Pool service account must have "modify" permissions.
- After saving a change to **Configuration>Session Recording**, the configurations for "Save Videos To Disk" and "Enable Deleting" will immediately be applied to all existing session recordings.

Using Network Share Path

In a clustered environment Secret Server needs to use a network path when saving the files to disk. All nodes need access to the path to read the videos back to the user.

To archive or save to a file path that is a network share, instead of a local folder:

- The SServer IIS application pool must be running as a service account. See "Running the IIS Application Pool As a Service Account" on page 60.
- You must grant access to the network share (using Windows ACLs) to the account running the Secret Server IIS application pool.

'

Installing the Advanced Session-Recording Agent

Session Recording Metadata Overview

By default, Secret Server session recording creates videos of launched sessions. Secret Server supports logging additional keystroke metadata for RDP and SSH sessions without requiring an agent across both Windows and Mac environments. When these options are enabled, users can search for keystrokes or applications across sessions and the session playback interface displays the additional information.
SSH metadata relies on the Secret Server SSH proxy.

Recording Remote Desktop session metadata requires the installation of an Advanced Session Recording Agent (ASRA) on the target server. In this scenario Secret Server's protocol handler is still used to launch the session and record the session video, and the ASRA records the metadata only.

ASRA can optionally record any session on the target server. If enabled and the session was not launched from Secret Server, ASRA will record video, keystrokes, and process metadata for the session and upload both to Secret Server once the session is disconnected. This works even if someone logs into the console directly or Remote Desktops into the server without using Secret Server at all. Live viewing of this type of session is not supported.

If you are licensed for session recording, you can install unlimited numbers of ASRAs.

How Advanced Session Recording Agents Work

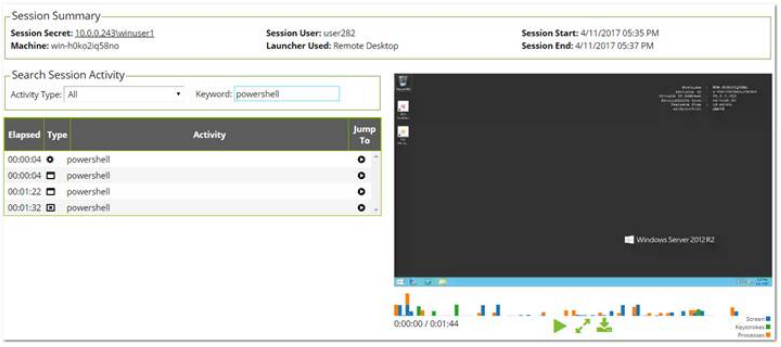
1. Once the ASRA is installed, it contacts the ASRA callback URL to determine if it should record metadata or video any time someone logs on to the computer.
2. A user logs on.
3. The ASRA sends Secret Server the computer's hostname, the username of the user who logged on, any domain name if available, and a list of the computer's IP addresses.

   > This data is not logged by Secret Server unless you enable DEBUG logging, only for troubleshooting purposes.

4. Secret Server checks for any recently-launched protocol handler sessions with matching details, and tells the ASRA if it should record the session.
   - If there is a match, the ASRA starts recording metadata and sends it back to Secret Server over the chosen response bus for the duration of the session.
   - If there is no match and "Record All Sessions" is enabled for the ASRA, the ASRA records both the video and metadata for the session.
   - If there is no match and "Record All Sessions" is not enabled, the ASRA records no video or metadata and waits for the next person to log in.
5. Once a recording session has been closed:
   - If the session was launched from Secret Server using the protocol handler, the video from the protocol handler is matched up to the metadata provided by the ASRA and combined.
   - If the session was not launched from Secret Server and "Record All Sessions" is enabled, the ASRA uploads both the video recording and the metadata to Secret Server.
6. On the Session Monitoring page, additional icons are presented based on what extra metadata is present for that session, such as keystroke data for both RDP and SSH, and process data for RDP.
7. Once the session recording has been processed, on the Session Playback page the additional metadata is visible:

   And on the Session Monitoring page:



In this example, we searched for activity where the user typed in the word “PowerShell.”

Record All Sessions

Recording video and metadata for all sessions on a server including console access requires Distributed Engine and ASR to be enabled as described above. ASRA must be installed on the target server, and it must be set to "Record All Sessions." If the server is set to “Only Record Secret Sessions,” the ASRA will only provide metadata when people launch into the server from Secret Server. Secret Server When recording all sessions, they appear on the Session Monitoring page. They are not tied to any specific secret because the sessions are not started by Secret Server.

Secret Server Configuration

First, session recording must be enabled (**Admin > Configuration > Session Recording**). As that page warns, Delinea highly recommends using RabbitMQ when using session recording in any production environments. See "Configuring Session Recording" on page 1234 for more information.

> For the highest scalability and reliability, Delinea recommends using RabbitMQ. MemoryMQ is an easier but less capable alternative and can be used for trials and proof of concepts but **should not** be used for production environments. Two exceptions are very small deployments and customers that do not use open-source software for compliance reasons.

SSH Metadata

To record SSH keystroke data, enable the SSH proxy (**Admin > SSH Proxy**). Individual secrets then require configuration of the Enable Proxy setting and the Enable Session Recording setting. Then when the SSH session is launched and recorded, keystroke data is recorded, which can be searched and is displayed in the session playback interface. See "Session Recording Overview" on page 1226 and "SSH Proxy Configuration" on page 831 for more information.

Remote Desktop Metadata

To record RDP session metadata, first distributed engine needs to be enabled (**Admin > Distributed Engine**) with an appropriate response bus site connector, which should be a RabbitMQ site connector in production environments. The ASRAs will communicate with the chosen site connector to return any recorded metadata. Next, the Advanced Session Recording feature must be enabled (**Admin > Configuration > Session Recording > Configure Advanced Session Recording**), and an ASRA callback URL entered. HTTPS should always be used in

production environments. Individual secrets then just need the Enable Session Recording setting enabled, and the computer you launch into must have the ASRA installed. The secrets do not have to use SSH proxy since the ASRA is what records the metadata. See "Session Recording Overview" on page 1226 for more information.

Session Recording Worker Role

> All cores are physical unless otherwise noted.

There is a dedicated Session Recording Worker role. If you have a clustered Secret Server environment, you can pick which nodes process recordings on the **Admin > Server Nodes** page. In a large environment with many recordings, you can configure nodes to be dedicated just to session recording, letting other nodes run the Background Worker and other roles. Session recording processes multiple videos at once, which can be controlled with the PrefetchCount.ConvertVideoMessage AppSetting (default: 2). We recommend setting this AppSetting to half the number of CPU cores on the server as a starting point. This setting applies only when using a RabbitMQ site connector, which is another reason Delinea highly recommends using Rabbit if you are using session recording.

Advanced Session Recording Agent

First, create one or more collections to group the ASRAs together, for example, for different domains or environments. Each collection has a unique installer that you can download from their page – the installer is customized to know which collection it is associated with. On the collection, you can specify if you want new agents to **Record All Sessions**, or to **Only Record Secret Sessions**. New agents adhere to this setting, and you can toggle it for individual agents, once they have registered.

Agent Manual Installation

The downloaded installer can be manually installed on a computer by running the setup.exe inside the zip file. It can also be deployed using group policy software installation or other MSI management software. The ASRA installs itself in `C:\Program Files\Thycotic Software Ltd\Session Recording Agent` and adds a Windows service, Delinea Session Recording Agent.

> Only 64-bit Windows operating systems are currently supported. .NET Core is also required.

Agent Updates

The ASRA does not automatically update, so new versions must be manually installed or re-deployed using the Group Policy MMC.

Agent Uninstallation

You can deactivate specific ASRAs or an entire collection in Secret Server, and the next time the ASRA reaches out to the ASRA callback URL, it will uninstall itself. Since it only reaches out when someone logs on to the computer, it will remain uninstalled until someone logs on again. The ASRA can also be manually uninstalled directly on the computer like any normal Windows application, but then Secret Server will still show it in the list of active ASRAs under its collection. It should also be deactivated in Secret Server to keep the agent list accurate. If "Record All Sessions" is enabled for this server, you will get a prompt to stop the Delinea Session Recorder application when

you attempt the uninstall because it is running and recording your session, as expected. To avoid these issues, we recommend using the Deactivate feature in Secret Server and then logging into the machine, and it will uninstall itself on its own.

Agent Group Policy Installation

Task 1: Review the Prerequisites

The ASRA requires a 64-bit operating system with .NET Core or greater installed on the client machine. This is the version that ships with Windows 8.1 and Windows 2012 R2.

Task 2: Download the Advanced Session Recording Agent Installer

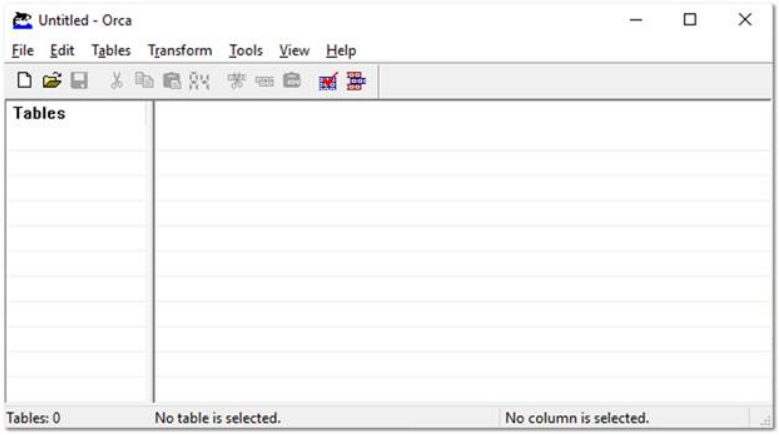
1. Log on to Secret Server.
2. Go to **Admin > Configuration > Session Recording > Configure Advanced Session Recording**.
3. Click on an existing collection, or create a new one, as appropriate.
4. Click the **Download Session Recording Installer** (64-bit) button. The installer is downloaded to your computer.

> The zip file is customized for each collection. Be sure to download the installer from the collection you want your new ASRAs associated with.

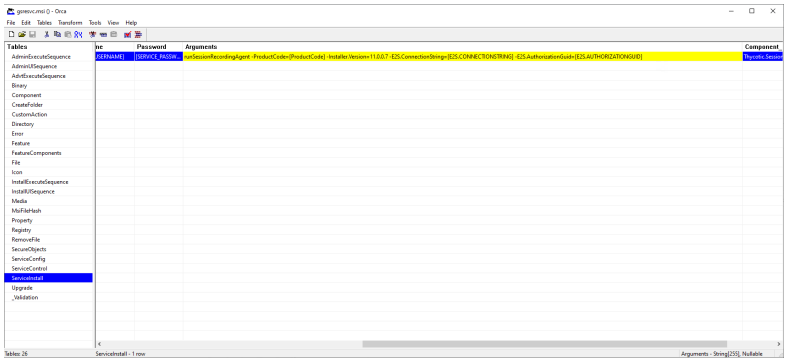
Task 3: Customize the Installer

For a normal manual installation, you extract the zip file, and run setup.exe. There are settings saved in setup.exe.config that customize the installation of the MSI file contained in the zip (gsresvc.msi). When you deploy the ASRA using Group Policy software installation instead of a manual one, the only other files in the zip you need is the MSI, which is deployed from a network share, and an MST (Master Software Tools?) “Transform” file which configures the custom settings.

1. Install Microsoft’s free MSI editing tool, Orca, if you do not already have it installed.
2. Extract the ASRA zip file into its own folder.
3. Right click on the MSI file (gsresvc.msi) in the folder where you extracted the zip and select **Properties** to verify that there is a **Digital Signatures** tab indicating that the MSI was signed by Delinea Software.
4. Launch Orca.



5. Open the extracted MSI file (gsresvc.msi). The Tables list appears.
6. Click the **Transform** menu at the top
7. Select **New Transform**.
8. In the **Tables** list, click **ServiceInstall**. Only one row should be listed on the right.



> This screen shot shows the Arguments column dragged wider to see its contents. When you initially see it, it will be very narrow, barely showing the contents.

9. Scroll to the **Arguments** column and copy and paste its contents into a text editor. It should look like this example. Be sure to select the entire column. You might need to adjust the column width.

> The entire string of text is essentially a CLI command with parameters that begin with a hyphen. For illustration purposes we put each parameter on its own line below.

```
                    runSessionRecordingAgent
-ProductCode=[ProductCode]
-Installer.Version=10.6.000000
-E2S.ConnectionString=[E2S.CONNECTIONSTRING]
-E2S.AuthorizationGuid=[E2S.AUTHORIZATIONGUID]
```

Everything highlighted is what we will customize. The fields in brackets are what setup.exe would normally customize.

10. In your text editor, open the setup.exe.config XML file from the zip. You will get the Globally Unique Identifier (GUID) from it.
11. In your text editor, replace each of these with the correct values as listed below. The GUID will require looking in setup.exe.config in the XML block. The values are as follows:
    - **ProductCode** should always be: "{A7FA0ADA-BEED-4841-9D3E-9D700B36F653}" (not in setup.exe.config).
    - **Installer.Version** should match your Secret Server version (visible in Secret Server in the bottom right corner).
    - **E2S.ConnectionString** is the callback URL for Secret Server Cloud is the Cloud URI.

      Secret Server Cloud does not list a Callback URL on the Advanced Session Recording page.

    - **E2S.AuthorizationGuid** is a unique GUID specific to the ASRA collection that you downloaded the installer file from. You can find it in the setup.exe.config file (in setup.exe.config). This is unique for each ASRA Collection.
12. Back in Orca, delete everything in the ServiceInstall Arguments column.
13. Copy and paste the customized version you just created from your text editor into the Arguments column.
14. Click the **Transform** menu.
15. Click **Generate Transform**.
16. Save the file as `gsresvc.mst` in the directory you extracted the installer into. This transform file now contains your customizations for the ServiceInstall Arguments.
17. Close Orca.
18. Check the MSI file's digital signature again to ensure that it was not edited: If you right-click the MSI file and select **Properties** again, the Digital Signatures tab should still show that the MSI is signed by Delinea Software. You created your own custom MST transform file, but the MSI itself should be unchanged. Orca can edit the MSI file itself, but that will invalidate Delinea's digital signature and it is unnecessary.

Task 4: Set up a Network Share

1. Place the `gsresvc.msi` and `gsresvc.mst` files on a network share on your domain controller.
2. Give "Authenticated Users" read access to this share.

   Computers in the domain will access this network share to get the installer files before any users log into the machine. It will be the machine account authenticating to the network share, before any users have logged in.

Task 5: Create a Group Policy with Software Installation to install the MSI

1. Open the Group Policy Management Console (**Start\ > Administrative Tools > Group Policy Management**).
2. Expand the **Forest** and **Domain** nodes until you locate the domain on which you are installing the ASRA.
3. Right click on **Group Policy Objects** and click **New**.
4. Enter a descriptive name for your GPO, such as " Delinea Session Recording Agent Installation," and click **OK**.
5. Right click on the newly created **GPO** node and click **Edit**.
6. Select **Computer Configuration > Policies > Software Settings > Software Installation**.
7. Right click on the **Software Installation** node and select **New > Package**.
8. Browse to the MSI on your network share using the share's UNC path, not its folder path. For example: `\\ServerMachineName\Shared and not C:\Shared`.
9. Click **Open**.
10. Click to select the **Advanced** option button.
11. Click **OK**. The name is automatically be set to "Delinea Session Recording Agent", since that is the product name in the MSI file.

    > You can customize the name here, but if you use something else, that is what you will want to check for in the Verify Configuration section, instead of "Delinea Session Recording Agent."

12. On the **Modifications** tab, click **Add** and select your MST transform file. Be sure to again use a UNC path like `\\ServerMachineName\Shared, not C:\Shared`.

    > If you wish to have the ASRA uninstalled when it falls out of management, click on the Deployment tab and click to select the box next to "Uninstall this application when it falls out of the scope of management".

13. Click OK.
14. In the group policy object editor, expand **Computer Configuration > Administrative Templates > System**.
15. Click the **Logon** node.
16. Right-click **Always wait for the network at computer start-up and logon** and select **Properties**.
17. Click **Enabled**.
18. Click **OK**. This helps reduce the number of reboots required for this policy to take effect as noted in the description of this option.

Task 6: Link your Group Policy Object to an OU

1. Open the Group Policy Management Console (**Start > Administrative Tools > Group Policy Management**)
2. Expand the **Forest** and **Domain** nodes until you locate the domain on which you are installing the Secret Server protocol handler.
3. Right-click the Organizational Unit (OU) for which you want Secret Server protocol handler to be installed and select **Link an Existing GPO**.
4. Select the GPO you created earlier.
5. Click **OK**. The GPO is now linked the entire OU.

> To immediately force the group policy change and install the software on a client machine, open a command console on the client machine (start > run > cmd), type `gpupdate /force`, and restart the client machine. You can also wait for the group policy to go into effect, which usually takes one to two hours, but a reboot will still be required due to the mechanics of group policy software installations.

Task 7: Verify Configuration at the Domain Level

1. Go to **Start > Administrative Tools > Active Directory Users and Computers**.
2. Right-click the OU for which Secret Server Protocol Handler is now configured and select **All Tasks > Resultant Set of Policy**.
3. Check to select the box next to **Skip to the final page of this wizard without collecting additional information**.
4. Click **Next** twice.
5. Click **Finish**.
6. In the new **Resultant Set of Policy** window, expand **Software Settings** under **Computer Configuration**.
7. Click to select **Software installation**.
8. **Delinea Session Recording Agent** should be visible in the **Installed Applications** column.

Task 8: Verify the Configuration of a Domain Member

1. From a command prompt, run `gpresult /h report.html` to output a report for just that one computer to the specified HTML file, which you can then view in a browser.
2. The Delinea session recording agent should be visible in the Installed Applications section.
3. Once the computer has rebooted and completed the installation, the software shows up in Apps and Features (Add Remove Programs). As usual, the Delinea Session Recording Agent Windows Service is installed in `C:\Program Files\Thycotic Software Ltd\Session Recording Agent`.

Stability and Compatibility with Older ASRAs

The latest Advanced Session Recording Agents (ASRAs) use more reliable durable message exchanges, which are not compatible with earlier (already deployed) ASRAs. Version 7.7+ of the ASRA only requires HTTP connectivity to Secret Server, the distributed engine response-bus site connector is no longer required.

To prevent this from breaking older Advanced Session Recordings, exchanges remain permanently transient. Newer ASRAs use HTTP uploads, which do not use the message queue. Older versions of ASRAs will continue to function as they have, while newer ASRA versions do not use the message queue and will have "durable" behavior over HTTP. We recommend updating your ASRA to version 7.7 or later as soon as possible.

Enabling Inactivity Timeout

If enabled, if a session appears idle, users are given a five-minute warning that they will be disconnected. A prompt appears that lets them choose to disconnect immediately or to continue the session. If no response is received, the session is disconnected five minutes later.

> This feature was added in Secret Server SP2 and is currently only supported in the Windows protocol handler (not Mac).

Enabling On-Demand Video Processing

As described above, this feature was added in Secret Server 10.6.24 to greatly improve session recording performance.

The Windows protocol handler now encodes the recording on the fly in WEBM format and streams the video to Secret Server. Once the session has ended, Secret Server reconstructs the video and leaves it in WEBM format, which Chrome and Firefox can natively play back.

Internet Explorer and Edge currently have issues playing back WEBM videos, so if you are using those browsers and try to view an on-demand recording, you are presented with a "Request Video Processing" button, which converts the video to H.264/MP4, as soon as possible, which IE or Edge can then play back.

If this option is not checked, all sessions recorded by the Windows protocol handler are converted to H.264/MP4 automatically. If you have many IE or Edge users, Delinea recommends leaving this option unchecked, but this will increase the processing time of videos and increase the load on your Secret Server servers that have the Session Recording role enabled.

This setting has no effect on sessions recorded with the Mac protocol handler, which is always encoded using your legacy video codec choice.

Recording Metadata

By default, session recording creates videos of the launched session. Secret Server supports logging additional metadata, such as keystrokes, for RDP and SSH sessions. When these options are enabled, users can search for keystrokes or applications across sessions, and the session playback interface shows additional activity information.

Session metadata requires Secret Server and the advanced session recording feature, which in turn requires an installation of an advanced session recording agent (ASRA) on the target servers. See "Installing the Advanced Session-Recording Agent" on page 1240.

SSH keystroke data relies on the Secret Server SSH Proxy or Session Connector (via the Protocol Handler). This can be enabled under Admin > SSH Proxy. See the "SSH Proxy Configuration" on page 831 for more information. Once proxying is enabled, recorded SSH sessions log SSH traffic, which can be searched and is displayed in the session playback interface.

Recording All Sessions

As of Secret Server SP2, you can configure the ASRA to record all sessions. This causes it to record video and metadata for anyone logging into the server, even when not using Secret Server, including logging into the console. Since these recordings are not tied to any specific secret, you must go to the **Admin > Session Monitoring** page to view them.

System Capacity Specifications

This applies to both ASR and basic session recording. See below for details.

Table: Session Recording Capacities

Web Node	Maximum Concurrent Session Conversions per Node	Maximum Processing Time per Session	Recording Processing Time per Maximum Length Session
Dedicated for session recording	4	2 hours	10 minutes
Shared for front-end processing and session recording	2	2 hours	20 minutes

See "Configuring the Maximum Concurrent Recording Sessions per Web Node" on page 1234.

Overview of Users, Roles, User Groups, and User Teams

Users

Users in Secret Server represent individual people, each with a unique username and other attributes. Users are assigned to groups, and roles are assigned to them either directly or via groups. This setup allows for granular control over what each user can access and perform within the system.

Roles and Role Permissions

Roles

Secret Server uses a role-based access control (RBAC) mechanism to regulate system access. Each user and group must be assigned to a role. Secret Server ships with three default roles: Administrator, User, and Read-Only User. Each role contains various permissions to match the job function of the user. Roles can be customized by assigning multiple permissions to a role, which can then be assigned to a user or group.

The Unlimited Administrator permission allows the user to have unlimited administrator rights when Unlimited Administrator is enabled in the configuration. By default, it is disabled.

To see the built-in roles and what permissions they possess, click the desired role link on the Admin > Roles page.

Role Permissions

Role permissions in Secret Server are granular and can be assigned to roles to control what actions users can perform. Some examples of role permissions include:

- **View Active Directory**: Allows a user to view, but not edit, Active Directory settings.
- **View Configuration**: Allows a user to view, but not edit, general configuration settings.
- **View Secret**: Allows a user to view which secrets exist in the system.
- **Administer Teams**: Allows a user to create, edit, and view all teams.

For a comprehensive list of role permissions, visit the "Secret Server Role Permissions List" on page 1257.

User Groups

User groups in Secret Server allow administrators to manage users collectively. Users can belong to different groups and inherit the sharing permissions and roles attributed to those groups. This simplifies the management of permissions and roles that can be assigned to a user. Groups can also be synchronized with Active Directory to further streamline management.

For a comparison of user groups and teams, see "User Teams Overview" on page 1284.

User Teams

User teams in Secret Server are special groups created to restrict what users can see. A team bundles users and groups to assign them the same rules regarding visibility of other users and sites. This is particularly useful for managed service providers or large companies that need to isolate users by department or customer.

Team-related permissions include:

- **Administer Teams**: Users can create, edit, and view all teams.
- **Unrestricted by Teams**: Users can view all users, groups, and sites, regardless of team affiliation.
- **View Teams**: Users can view all teams.

For a more in-depth overview, including a comparison of user groups and teams, see "User Teams Overview" on page 1284.

Overview of Users, Roles, User Groups, and User Teams

Users

Users in Secret Server represent individual people, each with a unique username and other attributes. Users are assigned to groups, and roles are assigned to them either directly or via groups. This setup allows for granular control over what each user can access and perform within the system.

Roles and Role Permissions

Roles

Secret Server uses a role-based access control (RBAC) mechanism to regulate system access. Each user and group must be assigned to a role. Secret Server ships with three default roles: Administrator, User, and Read-Only User. Each role contains various permissions to match the job function of the user. Roles can be customized by assigning multiple permissions to a role, which can then be assigned to a user or group.

> The Unlimited Administrator permission allows the user to have unlimited administrator rights when Unlimited Administrator is enabled in the configuration. By default, it is disabled.

> To see the built-in roles and what permissions they possess, click the desired role link on the Admin > Roles page.

Role Permissions

Role permissions in Secret Server are granular and can be assigned to roles to control what actions users can perform. Some examples of role permissions include:

- **View Active Directory**: Allows a user to view, but not edit, Active Directory settings.
- **View Configuration**: Allows a user to view, but not edit, general configuration settings.
- **View Secret**: Allows a user to view which secrets exist in the system.
- **Administer Teams**: Allows a user to create, edit, and view all teams.

> For a comprehensive list of role permissions, visit the "Secret Server Role Permissions List" on page 1257.

User Groups

User groups in Secret Server allow administrators to manage users collectively. Users can belong to different groups and inherit the sharing permissions and roles attributed to those groups. This simplifies the management of permissions and roles that can be assigned to a user. Groups can also be synchronized with Active Directory to further streamline management.

> For a comparison of user groups and teams, see "User Teams Overview" on page 1284.

User Teams

User teams in Secret Server are special groups created to restrict what users can see. A team bundles users and groups to assign them the same rules regarding visibility of other users and sites. This is particularly useful for managed service providers or large companies that need to isolate users by department or customer.

Team-related permissions include:

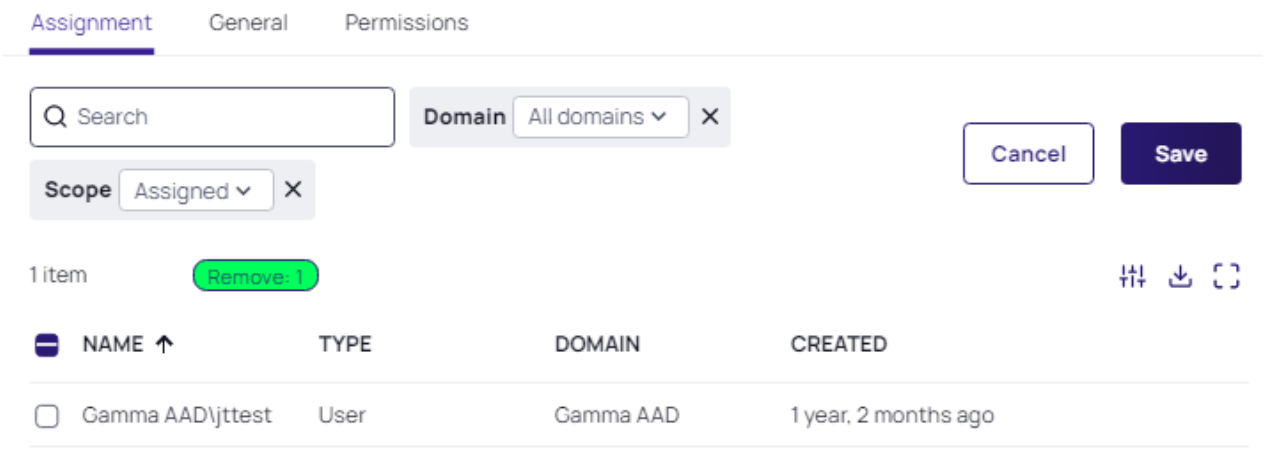
- **Administer Teams**: Users can create, edit, and view all teams.
- **Unrestricted by Teams**: Users can view all users, groups, and sites, regardless of team affiliation.
- **View Teams**: Users can view all teams.

For a more in-depth overview, including a comparison of user groups and teams, see "User Teams Overview" on page 1284.

Assigning Roles to a User

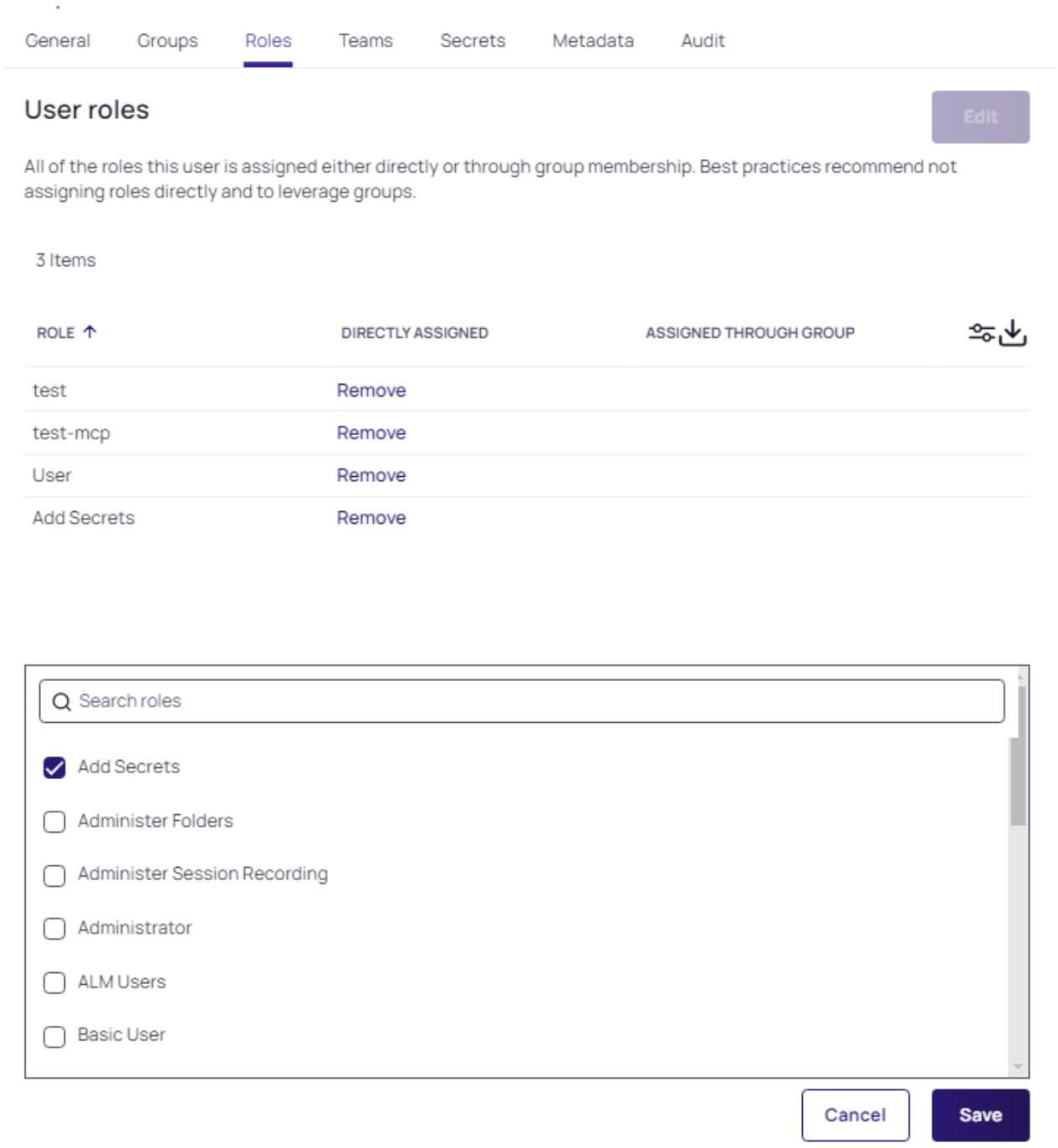
Choose one of three methods to assign roles to users or groups:

- **Via the Roles Assignment tab**
  1. Go to **Access > Roles**.
  2. Choose or create the role to assign.
  3. In the **Assignment** tab, click **Edit** to add or remove groups and users to this role.
  4. Change the **Scope** option to **Unassigned** to display the groups and users you can choose from.
  5. To remove a user or group, change the **Scope** option to **Assigned** and deselect the checkbox next to the user:



  6. Click **Save** to apply the changes.

- **Via the Users Roles tab**
  1. Go to **Access > Users**.
  2. Choose or create the user you wish to modify.
  3. In the **Roles** tab, click **Edit**.
  4. In the search bar, type the roles you wish to add or remove for this user.
  5. Select the checkbox next to each role you wish to add and **Save**:

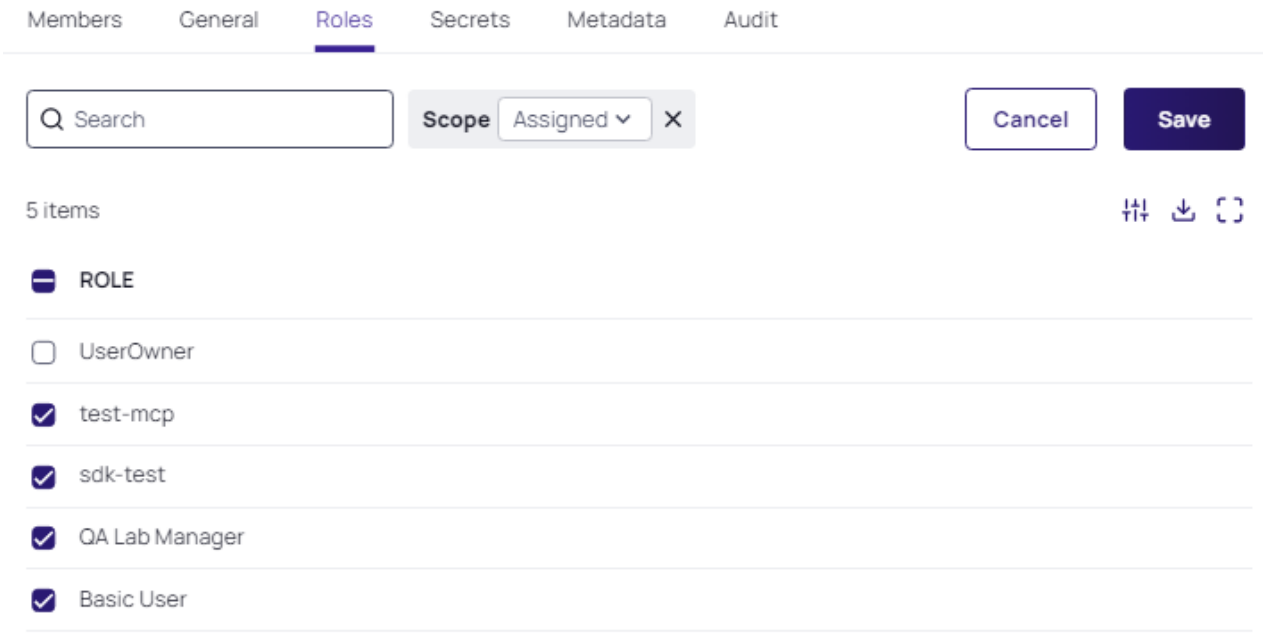


For the roles already assigned, deselect the checkbox next to their name in the search list. Alternatively, click **Remove** from the assigned role list to remove a role.

- **Via the Groups Roles tab**

1. To assign multiple roles to a single group, go to **Access > Groups**.
2. Click on the existing group you want to modify or create one.
3. In the **Roles** tab, click **Edit**.
4. Change the **Scope** option to **Unassigned**.
5. From the list that appears, select the checkbox for each role you wish to add.
6. Click **Save** to apply the changes.

   Optionally, change the **Scope** option to **Assigned** and deselect the checkbox next any role you wish to remove, before clicking **Save**:



Creating Roles

You can create roles from the Roles page. To get to the Roles page, navigate to **Administration > Roles**. Click the **Create Role** button to add the role.

Editing Role Permissions

To add or remove permissions to an existing role, click the role name of the role you wish to edit.

Select the permissions tab which lists all of the current role permissions that are currently assigned to this role. Add or remove role permissions by clicking edit. Once in edit mode you can remove roles by unchecking them when viewing "Assigned" or add new role permissions by checking them when viewing "Unassigned." Chips will indicate pending changes and once you are done modifying the role permissions click save.

Secret Server Role Permissions List

Overview

Secret Server uses role-based access control (RBAC) to regulate permissions. The roles are assigned to users or groups. A complete list of the permissions available to roles appears below:

To see the built-in roles and what permissions they possess, click the desired role link on the Admin > Roles page.

Adding a jumpbox route to a target secret: A user must have owner permissions on a secret to assign, change, or remove that secret's jump server route. Additionally, users are only able to pick from a list of routes where they have at least list permission on the first jump route server. Editing Jumpbox Routes: Users must have the "Administer Jumpbox Route" permission to create, edit, or deactivate jump server routes. Users with the "View Jumpbox Route" permission can view the details of all jump server routes in the Admin Jumpbox Route page, but they cannot make any changes.

Complete List

Access Offline Secrets on Mobile

Allows a user to cache their Secrets in the Secret Server mobile application for offline use. This permission does not automatically come with the Administrator role.

Add from External Directory

Add users from external directory. Cloud only.

Add Secret

Allows a user to create new Secrets. The Add permission no longer include the role permission View Secret.

Add Secret Custom Audit

Allows a user to make a custom audit entry when accessing a Secret using the web services API.

Administer Active Directory

Allows a user to view domains, edit existing domains, delete domains, and add new domains. Also allows a user to force synchronization or set the synchronization interval.

Administer Automatic Export

The user can do everything the other automatic export permissions allow *and* edit the automatic export configuration.

Administer Backup

Allows a user to view and configure automated backups for Secret Server. Users with this role permission can change the backup path, disable backups, and set the backup schedule. On-Premises only.

Administer Configuration

Allows a user to view and edit general configuration options. For example, a user with this role permission can turn on "Force HTTPS/SSL" and disable "Allow Remember Me".

Administer Configuration Proxying

Allows a user to view and edit SSH Proxy settings.

Administer Configuration SAML

Allows a user to view and edit SAML integration settings on the Login tab of Configuration settings.

Administer Configuration Security

Formerly "Administer Security Configuration," allows a user to view and edit security configuration options in Secret Server. Currently, these include enabling FIPS compliance mode and protecting the encryption key.

Administer Configuration Session Recording

Allows a user to view and edit session recording settings on the Session Recording tab of Configuration settings.

Administer Configuration Two Factor

Allows a user to change the configuration settings of the two factor authentication that are available for users logging into Secret Server.

Administer Configuration Unlimited Admin

Formerly "Administer Unlimited Admin Configuration," allows a user to turn on Unlimited Admin Mode. When this mode is enabled, users with the "Unlimited Administrator" role permission can view and edit all Secrets in the system, regardless of permissions. Note that you can assign "Administer Unlimited Admin Configuration" to one user and "Unlimited Administrator" to another user. This would require one user to turn on the mode and another user to view and edit secrets.

Administer ConnectWise Integration

Allows a user to view and edit configuration options for synchronizing with ConnectWise. This can be accessed through the "Folder Synchronization" link on the Administration page. Note that you need at least view access on the sync folder in order to set up or edit the ConnectWise integration.

Administer Create Application Accounts

Formerly "Create Application Account", allows a user to create application user accounts to be used exclusively for accessing Secret Server via the API. This permission allows for creating user accounts without the Administer Users or Administer Create Users permissions.

Administer Create Users

Allows a user to create new local users in Secret Server, but not edit them once created.

Administer Custom Password Requirements

Allows a user to view and edit custom password requirements that can be configured under the Security tab for individual Secrets.

Administer Data Retention

Can manage audit data retention, such as editing and running now. This permission does not automatically come with the Administrator role.

Administer DevOps Secrets Vault Tenants

Add, remove, and edit DSV tenants that automatically synchronize with Secret Server on a schedule.

Administer Disaster Recovery

Allows a user to configure instances as data sources or replicas for Disaster Recovery. Also allows user to initiate or test Data Replication and view related logs and audits.

Administer Discovery

Allows a user to view and import computers and accounts that are found by Discovery.

Administer Distributed Engine Configuration

Allows a user to update the Distributed Engine configuration.

Administer DoubleLock Keys

Allows a user to view, edit, create, and disable DoubleLock and QuantumLock keys. A DoubleLock or QuantumLock key acts as a separate encryption key to protect your most sensitive secrets. This option allows users to access and use the DoubleLocks/QuantumLocks link on the Administration page.

Administer Dual Control

Allows a user to view, edit, create, and disable Dual Control settings for reports and recorded sessions.

Administer Event Subscriptions

Allows a user to view, edit and create event subscriptions.

Administer Export

Allows a user to view the export log. Also allows users to export Secrets to which they have access to a clear text, CSV file.

Administer Folders

Allows a user to view, edit, create, move, and delete folders. Users still need the relevant view, edit, and owner permissions on the folders to perform these tasks.

Administer Groups

Allows a user to view, edit, create, and disable groups. Also allows users to assign users to groups and remove users from groups.

Administer HSM

Allows a user to change configuration or disable the use of a Hardware Security Module (HSM). On-Premise only.

Administer Inbox

Administer notification settings for the inbox.

Administer IP Addresses

Allows a user to create, edit, and delete IP Address Ranges. These ranges are used to restrict certain users to specific IP Addresses.

Administer Jumpbox Route

Allows a user to create, edit, or deactivate jump server routes.

Administer Key Management

Allows a user to enable, change, or disable the Key Management (Secret Server Cloud only).

Administer Languages

Allows a user to change the default language of Secret Server.

Administer Licenses

Allows a user to view, edit, install, and delete licenses.

Administer Lists

Allows a user to add, remove, and modify lists and list contents in Admin > Lists.

Administer Maintenance Mode

Allows a user to run maintenance mode.

Administer Metadata

Manage metadata fields and sections added to secrets and users in Secret Server.

Administer Nodes

Allows a user to view and edit server nodes and clustering settings. On-Premise only.

Administer OpenID Connect

Allows a user to manage OpenID connections.

Administer Password Requirements

Allows a user to view and edit character sets and password requirements.

Administer Pipelines

Allows a user to create, edit, and remove event pipelines and event pipeline policies.

Administer Platform Integration

Allows a user to manage the Secret Server connection to the Delinea platform.

Administer Remote Password Changing

Allows a user to turn Heartbeat and Remote Password Changing on and off globally. Also allows users to create new password changers and install password changing agents on remote machines.

Administer Reports

Allows a user to view, edit, delete, and create reports. Also allows users to customize report categories.

Administer Role Assignment

Allows a user to view which users and groups are assigned to which roles. Also allows users to assign users and groups to different roles.

Administer Role Permissions

Allows a user to view, edit, create and delete roles. Also allows users to assign different permissions to each role.

Administer Scripts

Allows a user to view and edit PowerShell, SQL, and SSH scripts on the Scripts Administration page.

Administer Search Indexer

Allows a user to view and edit search indexer options. These options control how searching in Secret Server works. For example, a user with this role permission could enable search indexing, which allows users to search on fields within a secret.

Administer Secret Policy

Allows a user to create and edit Secret Policies.'

Administer Secret Templates

Allows a user to view, edit, disable, and create Secret Templates.

Administer Security Analytics

Allows a user to view and edit the settings for Privilege Behavior Analytics.

Administer Session Monitoring

Allows a user to view and terminate active launcher sessions.

Administer SSH Cipher Suite

Allows a user to manipulate SSH cipher suite settings.

Administer SSH Menus

Allows a user to edit and create SSH Menus, used in allowlisting commands that can be used on a SSH session.

Administer System Log

Allows users to view and clear the System Log, which shows general diagnostics information for Secret Server.

Administer Teams

Users can create, delete, and view all teams.

Administer Template Custom Columns

Allows a user to enable the "Expose for Display" setting of a Secret template field to make it available for use in Dashboard custom columns.

Administer Users

Allows a user to create, disable, and edit users in the system.

This permission also allows a user to create and edit SDK/CLI rules.

Administer Workflows

Allows users to manage workflows (advanced access management).

Advanced Import

Allows a user to import Secrets from an XML file. Users with the this permission can import groups, folders, site connectors, sites, and secret templates, without having to create a secret. Users must have the Secret Server permissions needed for the objects listed in the XML.

Allow Access Challenge

Allows a user be challenged by Privileged Behavior Analytics if their behavior deviates from their normal behavior and meets certain requirements set by Privileged Behavior Analytics. Administrators do not have this permission by default.

Allow List Secret Access for Assigning Policy

This permission grants ability to assign a secret policy to a folder or secret if the user only has list access to the privileged account. This permission only applies to restrictions on the privileged accounts of the secret policy.

Without this permission, the user must have view access on the privileged account or will be unable to assign the secret policy.

> There are security concerns for assigning a policy with a privileged account the user does not have view access on since the policy will allow an internal threat actor to create a secret with this policy to gain access to resets by the privileged account, rendering them inherit access to the privileged account.

Approve via Duo Push

Allow a user to approve access requests via Duo push notifications. Administrators do not have this permission by default.

Assign Pipelines

Allows the user to assign an event pipeline policy to secret policies, or folders.

Assign Secret Policy

Allows a user to assign Secret Policies to folders and secrets.

Browse Reports

The "Browse Reports" role allows access to reports restricted by permissions. Permissions are configurable at the category and report levels and share a similar inheritance model to secrets and folders. You can define users or groups with "view" or "edit" permissions for each category or report.

> Users with the existing "view reports" and "edit reports" roles are not restricted by the permissions set.

Bypass Direct API Authentication Restriction

Allows users to ignore the PreventDirectApiAuthentication advanced setting and log in via the API with a non-application account.

Bypass SAML Login

Allows a user to login with local account without using SAML.

Copy Secret

Allows a user to copy secrets when that user also has Own Secret role permission.

Create Root Folders

Allows a user to create new folders at the root level of the folder structure.

Deactivate Secret

Allows a user to mark secrets as deactivated.

Deactivate Secrets from Reports

Allows a user to run the deactivate Secrets action from a report.

Download Automatic Export

The user can view all of the automatic export tabs *and* download exports from cloud storage (cloud customers only).

Edit Secret

Allows a user to edit secrets. Note that they still require the "Edit" or "Owner" permissions on the individual secrets they are editing.

Erase Secret

Allows a user to permanently erase (as opposed to deactivate, which is reversible) a secret.

Expire Secrets from Reports

Allows a user to expire Secrets listed in a report.'

Force Check In

Allows a user to force a secret that is checked out by another user to be checked in.

Migrate Data to Platform

This permission will be automatically applied to roles that contain both the Administer Users and Administer Platform Integration role permissions. This permission allows a user to migrate certain information to the connected Delinea Platform instance. Without a connected Delinea Platform instance the permission has no effect.

Own Group

Allows a user to be an owner of a group. This permission is in the default Group Owner role, which is automatically assigned when that user is set as owner of a group.

Own Secret

Formerly "Share Secret", allows a user to share secrets with other users. Also allows users to perform more advanced tasks on secrets of which they are "Owners", such as configuring expiration schedules, configuring the web launcher, converting secret template, and copying secrets (when a user also have the Copy Secret role permission.)

Own User

Allows the user to become a user owner, used to configure specific users without the Administer Users permission.

Personal Folders

Allows a user to have personal folder when the global personal folders configuration options is enabled.

Privilege Manager Administrator

Allows the user to have the "Administrator" role for Privilege Manager, giving full access to the system.

Privilege Manager Helpdesk User

Allows the user to have the "Help Desk" role for Privilege Manager, giving full access to approve or deny escalation requests.

Privilege Manager MacOS Admin

Allows the user to have the MacOS "Administrator" role for Privilege Manager, giving full access to the system.

Privilege Manager Unix/Linux Admin

Allows the user to have management permissions to Unix/Linux policies and machines.

Privilege Manager User

Allows the user to have the "User" role for Privilege Manager, giving read and write permissions to most items, but not rights to modify security permissions. Administrators do not have this permission by default.

Privilege Manager Windows Administrator

Allows the user to have the Windows "Administrator" role for Privilege Manager, giving full access to the system.

Rotate Encryption Keys

Allows a user to start a process that rotates the Secret encryption keys.

Run Automatic Export

The user can view all of the automatic export tabs and run the export manually by clicking the Run Export button.

Run Disaster Recovery Data Replication

Allows user to initiate or test Data Replication.

Run Scripts

Separates privileges in script management. Holders of the "View Scripts" role permission cannot execute test runs of scripts, and this permission must be assigned to perform this task.

Administer Scripts remains unchanged and allows view, edit, and run permissions.

Secret Launch

Dictates whether or not a user can launch a secret. Previously, a user could launch a secret if their user's role had the "View Secret" permission. As of Version 11.5, a user needs this permission to launch. A user will also need the "Secret Launch Remote Access (Platform)" permission to be able to launch a Remote Session with (RAS)

Secret Launch Remote Access (Platform)

Dictates whether or not a user can launch a secret. Previously, a user could launch a secret if their user's role had the "View Secret" permission. As of Version 11.5, a user needs this permission to launch a remote session with RAS.

Session Recording Auditor

Grants access to the session recording of a secret to a user with at least "List Access" permission on the secret. Administrators do not have this permission by default.

Users also need the "View Session Monitoring" permission to view the recordings in Secret Server.

Unlimited Administrator

Allows a user to view and edit all secrets in the system, regardless of permissions, when Unlimited Admin Mode is on. Note that another user with the "Administer Configuration Unlimited Admin" role permission would still need to turn this mode on. See "Unlimited Administration Mode" on page 273.

Unrestricted by Teams

Users can view all users, groups, and sites, regardless of team affiliation. Essentially, teams do not exist for the users with this permission, and the Teams page is not available to them. The default user role has this permission.

User Audit Expire Secrets

Allows a user to view the "User Audit" report, which shows all secrets that have been accessed by a particular user in a specified date range. Also allows the user to force expiration on all these secrets, which would make Secret Server automatically change the password.

View About

Allows a user to view the "About" page from the Help menu, which links to external resources such as Technical Support and the Delinea blog.

View Active Directory

Allows a user to view, but not edit, the Active Directory settings in the system.

View Advanced Dashboard

Allows a user to view advanced dashboard. Without this permission, users will only be able to view basic dashboard.

View Advanced Secret Options

Allows a user to view the Remote Password Changing, Security, and Dependency tabs on a Secret they have access to. Users must be able to successfully edit TOTP codes to view advanced secret options.

View All Session Recordings

Allows a user to view recorded sessions within Secret Server.

View Automatic Export

The user can view all of the automatic export tabs.

View Backup

Allows a user to view, but not edit, the automated backup settings. On-Premises only.

View Configuration

Allows a user to view, but not edit, general configuration settings.

View Configuration Proxying

Allows a user to view, but not edit, SSH Proxy settings.

View Configuration SAML

Allows a user to view SAML integration settings on the Login tab of Configuration settings.

View Configuration Security

Formerly "View Security Configuration," allows a user to view the security configuration of Secret Server.

View Configuration Session Recording

Allows a user to view session recording settings on the Session Recording tab of Configuration settings.

View Configuration Two Factor

Allows a user to view the configuration settings of the two factor authentication that are available for users logging into Secret Server.

View Configuration Unlimited Admin

Formerly "View Unlimited Admin Configuration," allows a user to view the Unlimited Admin Mode configuration. Also allows a user to view the Unlimited Admin Mode audit log.

View ConnectWise Integration

Allows a user to view, but not edit, the ConnectWise integration settings.

View Data Retention

Can view retained audit data. This permission does not automatically come with the Administrator role.

View DevOps Secrets Vault Tenants

View (not edit) the DSV tenants set to synchronize with Secret Server.

View Disaster Recovery

Allows a user to view configuration, logs and audits for Disaster Recovery.

View Discovery

Allows a user to view, but not edit, computers and accounts that are found by Discovery.

View Distributed Engine Configuration

Allows a user to view the Distributed Engine configuration.

View DoubleLock Keys

Allows a user to view which DoubleLock or QuantumLock keys exist in the system.

View Dual Control

Allows a user to view configured Dual Control settings for reports and Secret sessions.

View Event Subscriptions

Allows a user to view event subscriptions.

View Enterprise Objects

Allows a user to view user and secret metadata.

View Export

Allows a user to view the export log of the system to see when users exported secrets. Does not allow a user to export.

View Folders

Allows a user to view, but not edit, folders in the system.

View Group Roles

Allows a user to see which groups and users are assigned to which roles. Does not allow a user to change these assignments.

View Groups

Allows a user to see which groups exist in the system. Also allows a user to see which users belong to each group.

View HSM

Allows a user to view the Hardware Security Module (HSM) configuration settings. On-premises only.

View Inactive Secrets

Allows a user to view Secrets that have been deactivated in the system. This does not allow viewing of erased secrets, which are permanently gone.

View IP Addresses

Allows a user to view IP Address Ranges that have been created to restrict access. Does not allow a user to edit these ranges.

View Jumpbox Route

Allows a user to view the details of all jump server routes in the Admin Jumpbox Route page but not make any changes.

View Key Management

Allows a user to view the Key Management settings (Secret Server Cloud only).

View Launcher Password

Allows a user to unmask the password on the view screen of secrets with a launcher. Typically, this includes Web Passwords, Active Directory accounts, Local Windows accounts, and Linux accounts.

View Licenses

Allows a user to view, but not edit, the licenses in the system.

View Lists

View lists and list contents in Admin > Lists.

View Nodes

Allows a user to view, but not edit, the Secret Server web server nodes. On-premises only.

View OpenID Connect

View OpenID Connect integration settings in the Configuration Login tab. This replaces the Delinea One equivalent.

View Own Session Recordings

Restricts a user to only viewing the recordings that user initiated. If the user with this permission clicks on a recording initiated and owned by another user, he or she will get an Access Denied window.

View Password Requirements

Allows a user to view character sets and password requirements.

View Pipelines

Allows a user to view event pipeline policies and policy activities.

View Platform Integration

Allows a user to view the Secret Server connection to the Delinea platform.

View Remote Password Changing

Allows a user to view, but not edit, Heartbeat and Remote Password Changing settings.

View Reports

Allows a user to view, but not edit, reports. See "Browse Reports."

View Roles

Allows a user to view roles in the system. Also allows a user to see which groups are assigned to which roles.

View Scripts

Allows a user to view PowerShell, SQL, and SSH scripts on the Scripts Administration page.

View Search Indexer

Allows a user to view, but not edit, search indexer settings.

View Secret

Allows a user to only view which Secrets exist in the system.

Note: Prior to version 11.4, this controlled if a user could launch a secret. It has been supplanted with Secret Launch and Secret Launch Remote Access (Platform) for launching.

View Secret Audit

Allows a user to view Secret Audit.

View Secret Password and Private Key History

Allows a user to see the history of passwords, private keys, or passphrases in both old and new UI.

View Secret Policy

Allows a user to view, but not edit, Secret Policies.

View Secret Templates

Allows a user to view, but not edit, Secret Templates.

View Security Analytics

Allows a user to view, but not edit, settings for Privilege Behavior Analytics.

View Security Hardening Report

Allows a user to view the Security Hardening Report.

View Session Monitoring

Allows a user to view active launcher sessions.

View Session Recording Audit

Allow a user to view audits of recorded sessions.

View SSH Menus

Allows a user to view existing SSH Menus, used in allow-listing commands that can be used on a SSH session.

View SSH Cipher Suite

Allows a user to view SSH cipher suite settings.

View System Log

Allows a user to only view the System Log, which shows general diagnostics information for Secret Server.

View Teams

Users can view all teams. This is essentially a read-only Administer Teams.

View User Audit Report

Allows a user to view, but not edit, the User Audit Report.

View Users

Allows a user to view which users exist in the system.

View Workflows

View (not edit) workflows used for multi-tier secret-access approvals and secret erase requests.

Web Services Impersonate

Allows a user to send an approval request to act as another user within their organization when accessing Secret Server programmatically. Administrators do not have this permission by default.

Overview of Users, Roles, User Groups, and User Teams

Users

Users in Secret Server represent individual people, each with a unique username and other attributes. Users are assigned to groups, and roles are assigned to them either directly or via groups. This setup allows for granular control over what each user can access and perform within the system.

Roles and Role Permissions

Roles

Secret Server uses a role-based access control (RBAC) mechanism to regulate system access. Each user and group must be assigned to a role. Secret Server ships with three default roles: Administrator, User, and Read-Only User. Each role contains various permissions to match the job function of the user. Roles can be customized by assigning multiple permissions to a role, which can then be assigned to a user or group.

The Unlimited Administrator permission allows the user to have unlimited administrator rights when Unlimited Administrator is enabled in the configuration. By default, it is disabled.

To see the built-in roles and what permissions they possess, click the desired role link on the Admin > Roles page.

Role Permissions

Role permissions in Secret Server are granular and can be assigned to roles to control what actions users can perform. Some examples of role permissions include:

- **View Active Directory**: Allows a user to view, but not edit, Active Directory settings.
- **View Configuration**: Allows a user to view, but not edit, general configuration settings.
- **View Secret**: Allows a user to view which secrets exist in the system.
- **Administer Teams**: Allows a user to create, edit, and view all teams.

For a comprehensive list of role permissions, visit the "Secret Server Role Permissions List" on page 1257.

User Groups

User groups in Secret Server allow administrators to manage users collectively. Users can belong to different groups and inherit the sharing permissions and roles attributed to those groups. This simplifies the management of permissions and roles that can be assigned to a user. Groups can also be synchronized with Active Directory to further streamline management.

For a comparison of user groups and teams, see "User Teams Overview" on page 1284.

User Teams

User teams in Secret Server are special groups created to restrict what users can see. A team bundles users and groups to assign them the same rules regarding visibility of other users and sites. This is particularly useful for managed service providers or large companies that need to isolate users by department or customer.

Team-related permissions include:

- **Administer Teams**: Users can create, edit, and view all teams.
- **Unrestricted by Teams**: Users can view all users, groups, and sites, regardless of team affiliation.
- **View Teams**: Users can view all teams.

> For a more in-depth overview, including a comparison of user groups and teams, see "User Teams Overview" on page 1284.

Bulk Operations on Users

Bulk operations on users can also be performed from the **Users** page. Select one or more users using the check boxes beside the **Username** column, or select all or none by toggling the check box in the header row. Once the appropriate users have been selected, use the Bulk Operation list at the top of the grid to select an action. Bulk operations on users currently include enabling or disabling user access, as well as configuring users for email or RADIUS two-factor authentication.

Configuring Users

User settings can be modified by clicking the username in the **Username** column on the **Users** page. Search for users using the search bar at the top of the grid. To show users that are marked inactive, search for users using the search bar at the top of the grid. You can filter your search by domain, search text and active status.

Creating Users

To manually create a single user, navigate to **Admin > User Management** and select the **Create User** button. A dialog will open and you can enter the relevant information for a user.

> To add many users from your Active Directory setup, you can use Active Directory synchronization (see "AD and Secret Server Overview" on page 491).

Deleting Users

You cannot delete users per se because of auditing requirements; however, deactivating the user from the "User Settings" on page 1281 accomplishes the same thing. See "Removing Deactivated User PII" on the next page for eliminating all traces of a deactivated user.

Password Settings

The following settings are found in the **Administration > Configuration** page, inside the **Local User Passwords** tab. These settings apply to users that were created manually, not users brought into Secret Server through Active Directory synchronization:

- **Allow Users to Reset Forgotten Passwords:** If enabled, the "Forgot your password?" link appears on all users' login screens. Clicking on this link prompts the user to enter the email address that is associated with the user's Secret Server account. If the email address is found, then an email containing a link for password reset is sent. Note that this only works for local user accounts and not for Active Directory accounts. Secret Server

- **Enable Local User Password Expiration:** When enabled, Secret Server forces a password change for a user after a set interval. After the interval time has elapsed, the next time the user attempts to log in, they are prompted for the old password, a new password, and a confirmation of the new password. The new password is validated against all the password requirements. Newly created local users are also be prompted to change their password upon logging into Secret Server for the first time when this setting is enabled.
- **Enable Local User Password History:** If enabled, this prevents a user from reusing a password. For example, if set to "20 Passwords", this would prevent the user from using a password they have used the previous 20 times. This in conjunction with "Enable Minimum Local Password Age" helps ensure that users are using a new and unique password frequently rather than recycling old passwords.
- **Enable Minimum Local User Password Age:** If enabled, the value for this setting reflects the minimum amount of time that needs to elapse before a password can be changed. This prevents a user from changing their password too frequently, which allows them to quickly re-use old passwords.
- **Local User Password is valid for:** If enabled, this is the interval that a local user password is valid before it must be changed (see "Enable Local User Password Expiration" setting for details). If this setting is disabled, the entered value displays as "Unlimited".
- **Lowercase Letters Required for Passwords:** Force all user Secret Server login passwords to contain at least one lowercase letter.
- **Minimum Password Length:** Force all user Secret Server login passwords to contain a set minimum number of characters.

  The maximum number of characters is 1024.

- **Numbers Required for Passwords:** Force all user Secret Server login passwords to contain at least one number.
- **Symbols Required for Passwords:** Force all user Secret Server login passwords to contain at least one symbol, such as `!@#$%^&*`.
- **Uppercase Letters Required for Passwords:** Force all user Secret Server login passwords to contain at least one uppercase letter.

Removing Deactivated User PII

Overview

General Data Protection Regulation (GDPR) adherence raises the possibility that Secret Server users may make a data removal claim against a Secret Server administrator. This requires removing any personally identifiable information (PII) in Secret Server for that individual.

To address this, Secret Server has a button that automatically removes most PII for any deactivated user.

Removing the PII

1. Remove the user from Active Directory (AD). See Active Directory Considerations below.
2. In Secret Server, go to **Administration > User Management** and select the **Users** tab.
3. Click the user name link for the desired user. The View User Page appears.

4. Click the **Remove Personally Identifiable Information** button. A confirmation dialog box appears.

   > Once you confirm, the user cannot log on to Secret Server. Click the Cancel button if you are not positive this is what you want to do.

   Clicking the **OK** button will change these to random values or set them to null:

   - Username
   - Display name
   - Password
   - Personal folder name
   - Personal group name
   - RADIUS username

   In addition:

   - The user's AD GUID is cleared
   - The user's email address is removed from their record
   - The user's name is replaced with "<redacted>" in event audits where it can be clearly identified.
   - The PII removal is recorded in the user's audit

5. Click the **OK** button. The removal begins. Once complete, the Remove PII button disappears for that user.
6. (Optional) Run a query that scans the entire Secret Server database for the removed strings. You may want to do this because the process cannot find *all* potential instances of USER PII throughout Secret Server, such as that in secret names or notes.

> You can create an Event Subscription to "remove user PII" events.

Active Directory Considerations

We recommend removing the user from AD before removing the PII. If you remove the PII without first removing the user from AD, the user is reintroduced into Secret Server on subsequent AD synchs. This creates a new user account in Secret Server, which might require you to disable this new user account and remove its PII too (after removing the AD user).

Sorting and Searching for Users

Parameters

You can sort and sometimes search users based on a number of parameters, including:

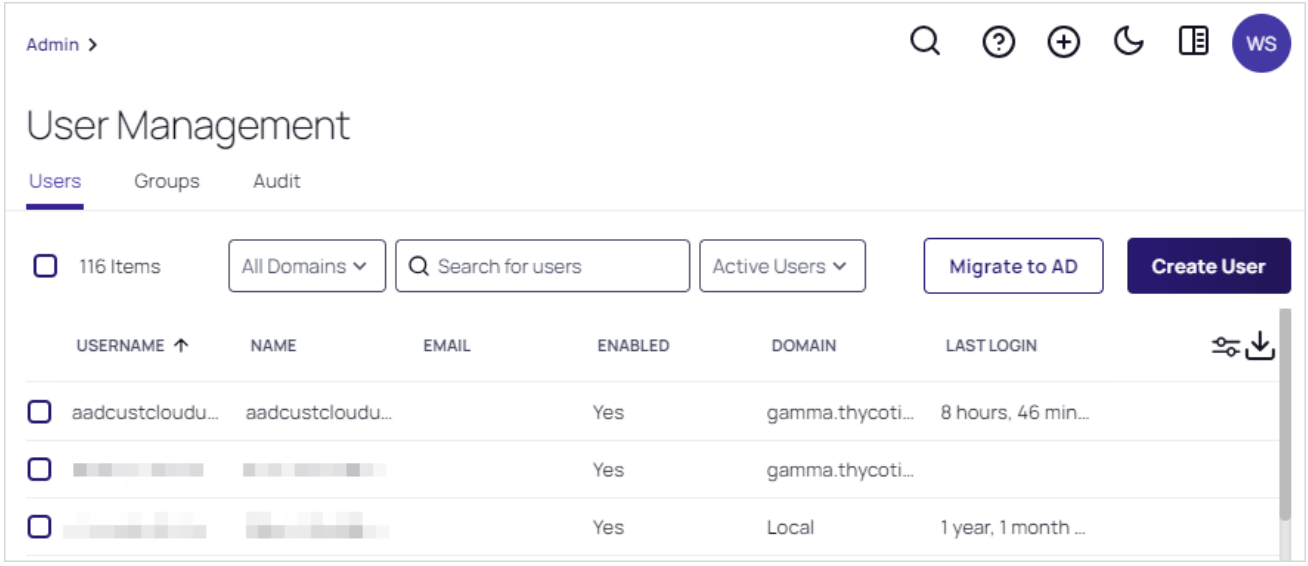
- Application account
- Created date
- Domain

- Email address
- Enabled status
- Last login date
- Locked out status
- Login failures
- Name
- OpenID Connect enabled status
- Platform integration status
- Two factor authentication method
- Username

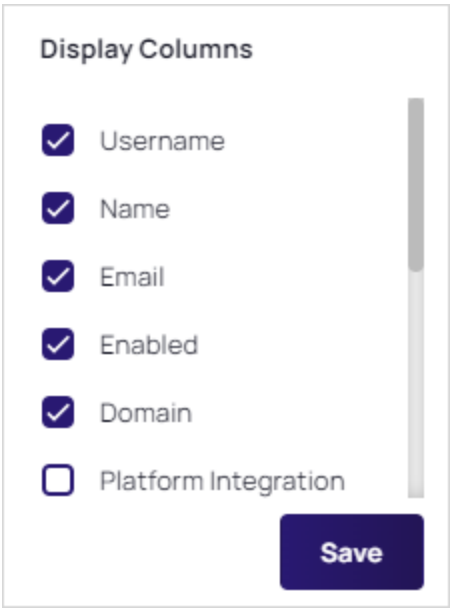
Viewing or Hiding Columns

Each of these items can appear in a column in the Users table. To enable or disable a column:

1. Go to **Admin > User Management**.
2. Ensure the **Users** tab is selected:



3. Click the icon. The Display Columns popup appears:



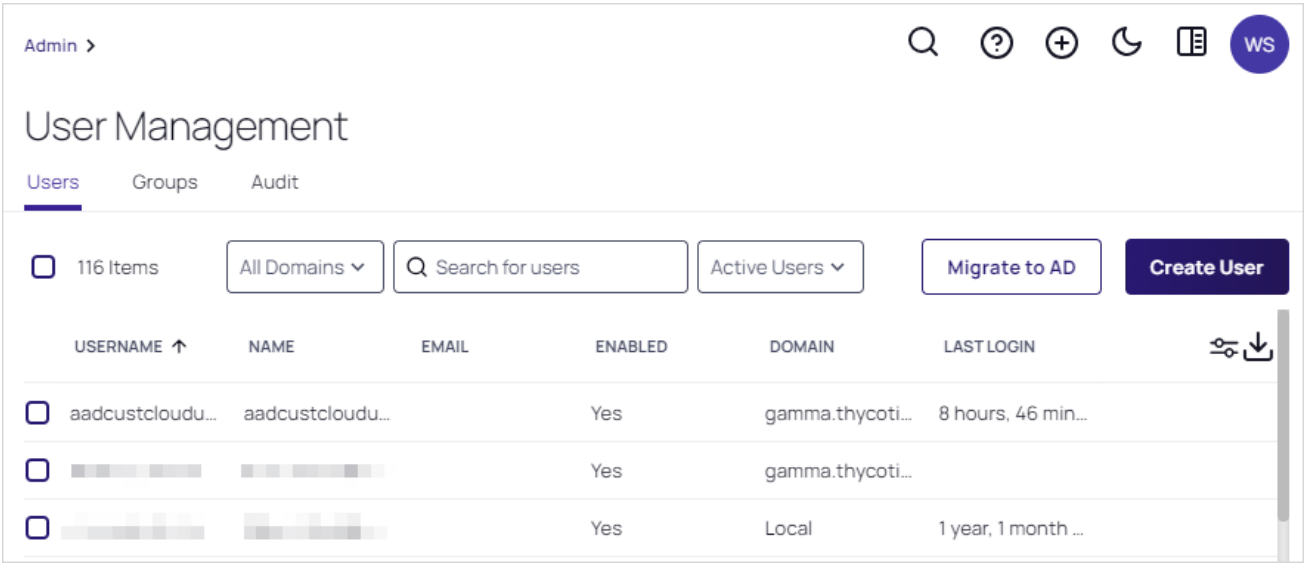
4. Click to select or deselect the columns you want to appear in the users table.
5. Click the **Save** button. The table updates to reflect your choices.

Sorting by Columns

To sort by a column, click the column heading you wish to sort by. Click the column heading again to reverse the sort order.

Searching for Users

1. Go to **Admin > User Management**.
2. Ensure the **Users** tab is selected:



3. Click the **Domains** dropdown list to limit the search to a listed domain.
4. Click the **Active Users** dropdown list to limit the search to active or inactive users.
5. Type the desired term to search for in the **Search** text box. You can search by:
   - Email address
   - Name
   - Username
6. Press **<Enter>**. The search results appear.

Unlocking Local Accounts

If a user fails their login too many times (specified in the **Local User Passwords** section of the configuration page), their account is locked out and they are not be able to log in.

To unlock the account:

1. Log on as an administrator.
2. Click on **Administration > User Management** and select the **Users** tab.
3. Click on the user who is locked out.
4. Click **Edit**.
5. Click to deselect the **Locked out** check box.
6. Click **Save**.

User Login Settings

Secret Server users can be set up with many different login requirements. For example, you can force strong Login passwords by requiring a minimum length and the use of various character sets.

The following settings are available under the **Administration > Configuration** page, inside the **Login** tab:

> "Allow Autocomplete" is now permanently on.

- **Allow Remember Me:** This option enables the Remember Me checkbox on the Login screen. When a user chooses to use remember me, an encrypted cookie is set in their browser. This enables the user to revisit Secret Server without the need to log in. This cookie is no longer be valid when the remember me period has expired. They then have to enter their login information again. This option allows users to remain logged in for up to a specific period (specified in the "Remember Me Is Valid for" setting mentioned below). This option can be a security concern as it does not require re-entry of credentials to gain access to Secret Server.

> "Remember me" is only visible if the "Allow Remember Me" setting is enabled. This is the period that the remember me cookie mentioned above is valid. For example: if set to one day, then users taking advantage of "remember me" have to log in at least once a day. To set a time value (minutes, hours, or days), uncheck the Unlimited checkbox.

- **Enable RADIUS Integration:** Allow for RADIUS server integration with your user login authentication. Other RADIUS settings appear upon enabling this option. These settings are discussed in "RADIUS User Authentication" on page 439.
- **Maximum Concurrent Logins Per User:** This setting allows a user to log into Secret Server from multiple locations at once without logging out their sessions at other locations.
- **Maximum Login Failures:** Set the number of login attempts allowed before a user is locked out of their account. Once locked out, they need a Secret Server administrator to reset their password and enable their account. For details on how to reset a locked account, see "Creating Users" on page 1273.
- **Require Two Factor for these Login Types:** This setting specifies which types of login require two-factor authentication:
  - Website and Web service Log on
  - Website log on only
  - Web service log on only
- **Visual Encrypted Keyboard Enabled:** This setting enables a visual keyboard for logins.
- **Visual Encrypted Keyboard Required:** This setting requires a visual keyboard for logins.

User Owners

User Administrators can also set another group or user as the *user owner* for a Secret Server local user. User owners can manage and edit just that user. For example, a developer might need to unlock or reset the password for an application account but should not have access to all users. Set **Managed by to User Owners** on a user and then select **Groups** or **Users**. Note that Unlimited Administrator mode can still be used to manage groups with user owners assigned.

User Preferences

Users can set their preferences by clicking on their profile icon in the top right and selecting User Preferences.

General TabThe following settings are available for users under the General tab:

- **API Token:** allows creating API Tokens for use in scripts. Click on the **Generate API Token and Copy to Clipboard** link to generate a token and copy it to clipboard automatically.
- **General Information:** contains general information about your user account within your organization's framework. The information in this section cannot be edited on this page. To update your display name, username, or email, please contact your administrator.
- **Password Settings** allows changing your local user password used to log in to Secret Server, or making changes to your user account's DoubleLock password. Click **Create DoubleLock Password**, in the pop-up enter and confirm a new password, and click **Create**.

Settings Tab

The following configuration settings are available for users under the Settings tab:

Preferences: personal user preferences associated to your account that control how the application looks, works, and displays data.

- **Color Mode:** select Light or Dark mode.
- **Communication Time Zone:** select time zone that will be used for dates when receiving communications.
- **Date Format:** select date format displayed for a user in Secret Server.
- **Language:** select language. Your browser language is set by default. Currently the following languages are supported: Chinese Simplified, Chinese Traditional, English, French, German, Japanese, Korean.
- **Login Home:** select the default home page after logging in if no redirect URL is specified.
- **Time Format:** select time format displayed for a user in Secret Server.

Email Settings: select what you would like to be notified about, based on the Secret events. Set the toggles to on or off to enable or disable the related settings.

- **Send Email When Dependencies Fail to Update:** enables emails to be sent when dependencies fail to update.
- **Send Email When Secrets are Changed:** enables emails to be sent on all changes of any secret that the user has view permission. There is a limit of one mail per five minutes per edit of the same user. For example, if user "User1" edits the secret twice within this grace period, only one email is sent.
- **Send Email When Heartbeat Fails for Secrets:** when enabled, the user is emailed when a heartbeat fails for any secret the user has view permission to.
- **Send Email When Secrets are Viewed:** enables emails to be sent on all views of any secret that the user has view permission. There is a limit of one email per five minutes per view of the same user. For example, if user "User1" views the secret twice within this grace period, only one email is sent.

Launcher Settings: allow setting launchers to fit the workstation needs. The settings will apply to all launchers you use where applicable. Set the toggles to on or off to enable or disable the related settings.

- **Allow Access to Printers:** enable access to printers when using the launcher.
- **Allow Access to Drivers:** enable access to drivers when using the launcher.
- **Allow Access to Clipboard:** enable access to clipboard when using the launcher.
- **Allow Access to Smart Cards:** enable access to smart cards when using the launcher.
- **Connect to Console:** allows connecting to remote machines using the Remote Desktop launcher and connecting as an administrator. This is the equivalent of using the `/admin` or `/console` switch when launching Remote Desktop.
- **Use Custom Window Size:** enable displaying Width and Height text boxes for the user to specify a custom window size for an RDP launcher.

Security Tab

User Sessions listed under the Security tab provide a record of known login sessions from this user account. User Sessions can be terminated which will perform a log out upon that session. To terminate a session, select it from the list, and click Terminate in the pop-up.

User Restriction Settings

The following restriction settings are available:

- **Enable Login Policy:** If enabled, this simply displays the policy. To force the acceptance of the policy.
- **Force Inactivity Timeout:** This setting is the time limit on idle Secret Server sessions. Once a session expires, the user must login again with their username and password.
- **Force Login Policy:** This setting forces the checking of the "I accept these terms" checkbox before allowing the user to login to SS.
- **IP Restrictions:** This setting can be entered by going to **Administration > IP Addresses**. In there, you can enter the IP ranges you wish your users to use. To configure a user to use the ranges, navigate to the **User View** page and click the **Change IP Restrictions** button. In the subsequent page, you can add all the ranges you want your user to use.
- **Login Policy Agreement:** The Login Policy Agreement is displayed on the login screen. You can change the contents of the Login Policy Statement by editing the file `policy.txt`. By default, this is not enabled. The settings to enable this are accessed by first navigating to **Administration > Configuration** and going into the **Login** tab. Then click the **Login Policy Agreement** button.

User Settings

Below is a brief explanation of each text-entry field or control:

- **Display Name:** Text that is used throughout the user interface, such as in audits.
- **Domain:** If a drop-down list is visible, selecting a domain from the list is one way to set the expected domain of the user. However, a more dynamic way to have this text-entry field (and all the other text-entry fields) set is through Active Directory synchronization.
- **Email Address:** Email address used for Request Access, email two-factor authentication, and the like.
- **Email Two-Factor Authentication:** On a login attempt, the user has an email sent to the email address entered above. This email contains a pin code that the user needs to log into the account. See "Email Two-Factor Authentication" on page 435 for details.
- **Enabled:** Disabling this control removes the user from the system. Effectively, this is the way to delete a user. Secret Server does not allow complete deletion of users due to auditing requirements. To re-enable a user, navigate to the **Administration > Users page**, check the **Show Inactive Users** checkbox just under the **Users** grid, and edit the user to mark them enabled (see "Configuring Users" on page 1273).
- **Locked Out:** If checked, then this user has been locked out of the system due to too many login failures. To remove the lock, uncheck the check box. For more details on locking out users, see Maximum Login Failures setting described in the Login Settings section.
- **Password:** Login password for the user. For the various login settings, see Login Settings section.
- **RADIUS Two-Factor Authentication:** This text-entry field only appears if RADIUS authentication is enabled in the configuration. On a login attempt, the user must enter the RADIUS token sent from the RADIUS server. See "RADIUS User Authentication" on page 439.

- **RADIUS User Name:** This text-entry field only appears if the above RADIUS Two Factor Authentication setting is enabled. This is the username the RADIUS server is expecting. See "RADIUS User Authentication" on page 439.
- **User Name:** Login name for the user.

> A new user is assigned the User role by default. For more information on roles, see "Roles."

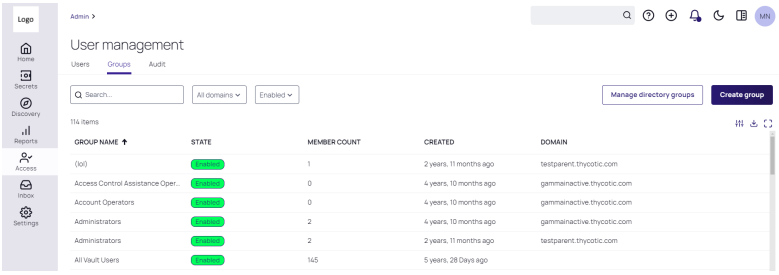
User Groups

Secret Server allows administrators to manage users through *user groups*. Users can belong to different groups and receive the sharing permissions, as well as roles, attributed to those groups. This setup simplifies the management of the permissions and roles that can be assigned to a user. Additionally, groups can be synchronized with Active Directory to further simplify management.

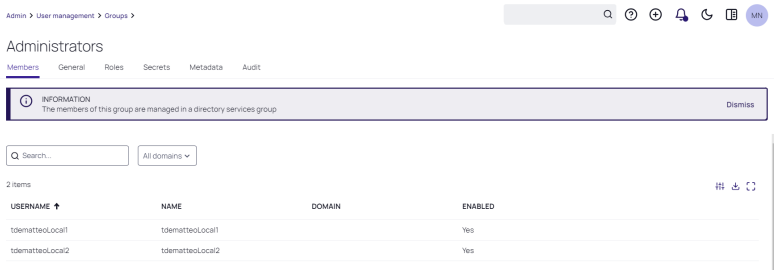
Assigning Group Owners

Group Administrators can also set another group or user as the group owners for a Secret Server local group. Group owners can manage membership just for that group. To assign the group owner:

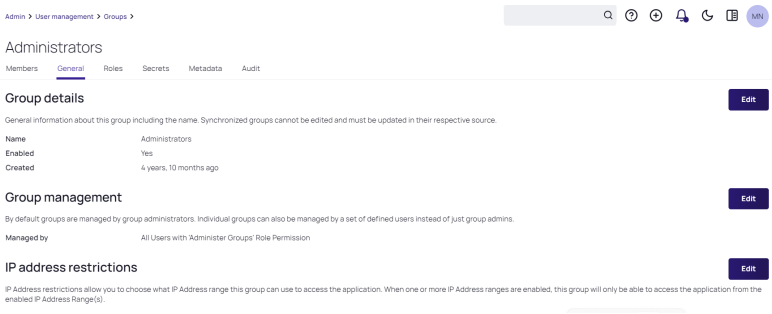
1. Navigate to **Admin** > **Groups**:



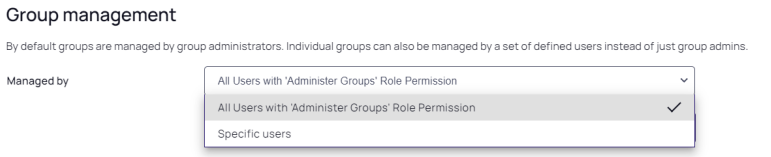
2. Click the desired group in the list. The Group's page appears:



3. Under the **General** tab, click **Edit** in the **Group management** section.



4. Click the **Managed By** dropdown list to select the owner.



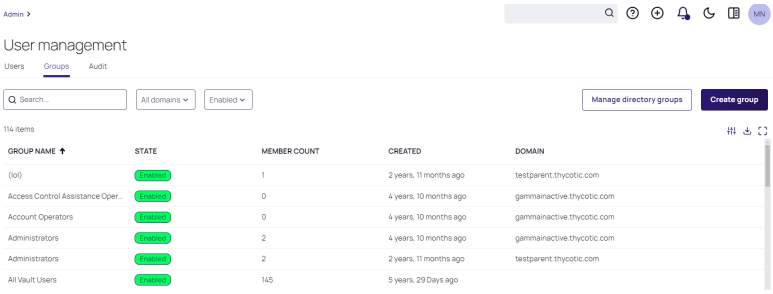
5. Click **Save**.

> Very commonly, the group owner is managed by Active Directory, not Secret Server:

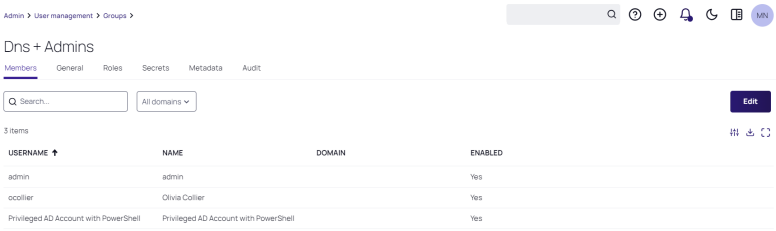
Assigning Users to Groups

On the Group Assignment page, users can be added and removed from the group.

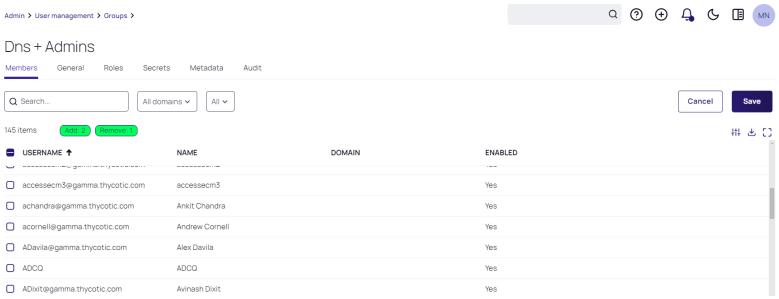
1. Navigate to **Admin** > **Groups**:



2. Select a group and click **Edit**.



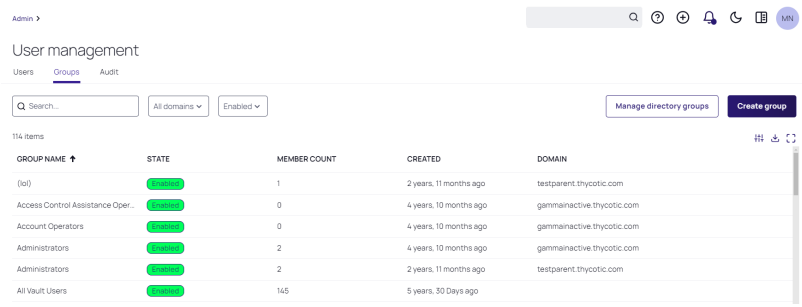
3. Select the related user domains (or all domains) from the dropdown to be displayed in the list. Select Assigned, Unassigned, or All users to be displayed in the list. Check to select users to be added, or uncheck users to be removed from the group. When done, click **Save**.



> If the group was created using Active Directory synchronization, this group is not be editable. See "AD and Secret Server Overview" on page 491.

Creating User Groups

You can create and edit groups from the Groups page. You can get to the Groups page by navigating to **Admin > Groups**



By either selecting an already existing group from the list, or clicking **Create Group**, you can modify or add the group.

> To add groups and the users inside them from your Active Directory setup, you can use Active Directory synchronization (see "AD and Secret Server Overview" on page 491).

User Teams Overview

Purpose of User Teams

With Secret Server teams, administrators can create special groups called *teams* to restrict what users can see. A team bundles users and groups to assign them the same rules as to what other users and sites are visible to them. For example, a managed service provider could isolate their customers from seeing other customer's user accounts or a large company could "firewall" their users by department. Site visibility can also be restricted by teams.

Teams are designed for shared secrets and do not apply to Secret Server administration as a whole.

Users *without* any team-related permissions are subject to team restrictions. The Unrestricted by Teams permission must be present to remove them. That is why the User role comes with that permission by default. See Team-Related Permissions.

Team restrictions are designed for regular users so granting additional administrative permissions can override the restriction. This applies to group owners, so if a user is assigned as a group owner, that user will be able to see all users when assigning members.

Team-Related Permissions

Team visibility and management are controlled by user roles. Those roles, and by extension users, are governed by the following team-related role permissions:

- **Administer Teams:** Users can create, edit, and view all teams.
- **No Teams-related Permissions:** Users can only view other users within their team.
- **Unrestricted by Teams:** Users can view all users, groups, and sites, regardless of Team affiliation. Essentially, teams do not exist for the users with this permission, and the Teams page is not available to them. The default user role has this permission.
- **View Teams:** Users can view all teams. This is essentially a read-only Administer Teams.

User Teams Versus User Groups

User teams and user groups in Secret Server serve different purposes and offer distinct functions. Here is a detailed comparison to highlight their differences and why user teams cannot be fully replaced by user groups and vice versa:

User Groups

- **Purpose**: User groups are primarily used to manage users collectively and simplify the assignment of permissions and roles.
- **Functionality**:
  - Users can belong to multiple groups.
  - Groups can be synchronized with Active Directory.
  - Permissions and roles assigned to a group are inherited by all users within that group.
- **Use Case**: Ideal for managing permissions and roles for a set of users who need similar access rights and capabilities within Secret Server.

User Teams

- **Purpose**: User teams are designed to restrict what users can see, particularly useful for isolating visibility of users and sites.
- **Functionality**:
  - Teams bundle users and groups to assign them the same visibility rules.
  - Teams can restrict site visibility, ensuring that users in one team cannot see users or sites assigned to another team.
  - Team-related permissions control visibility and management of teams.
- **Use Case**: Ideal for scenarios where visibility needs to be restricted, such as isolating departments within a large company or separating customers in a managed service provider setup.

Key Differences

Visibility Control

- **User Groups**: Primarily manage permissions and roles but do not inherently restrict visibility of other users or sites.
- **User Teams**: Specifically designed to control and restrict visibility of users and sites, ensuring isolation between different teams.

Granularity

- **User Groups**: Focus on collective management of permissions and roles.
- **User Teams**: Focus on visibility restrictions, which can be more granular and specific to organizational needs.

Use Case Suitability

- **User Groups**: Suitable for managing access rights and permissions across users who need similar capabilities.
- **User Teams**: Suitable for scenarios requiring strict visibility controls and isolation between different sets of users.

Conclusion

While user groups are effective for managing permissions and roles, they do not offer the same level of visibility control as user teams. User teams provide a more granular approach to restricting what users can see, which is essential for certain organizational structures and security requirements.

Configuring Teams Management

To set up Secret Server to use the team management feature:

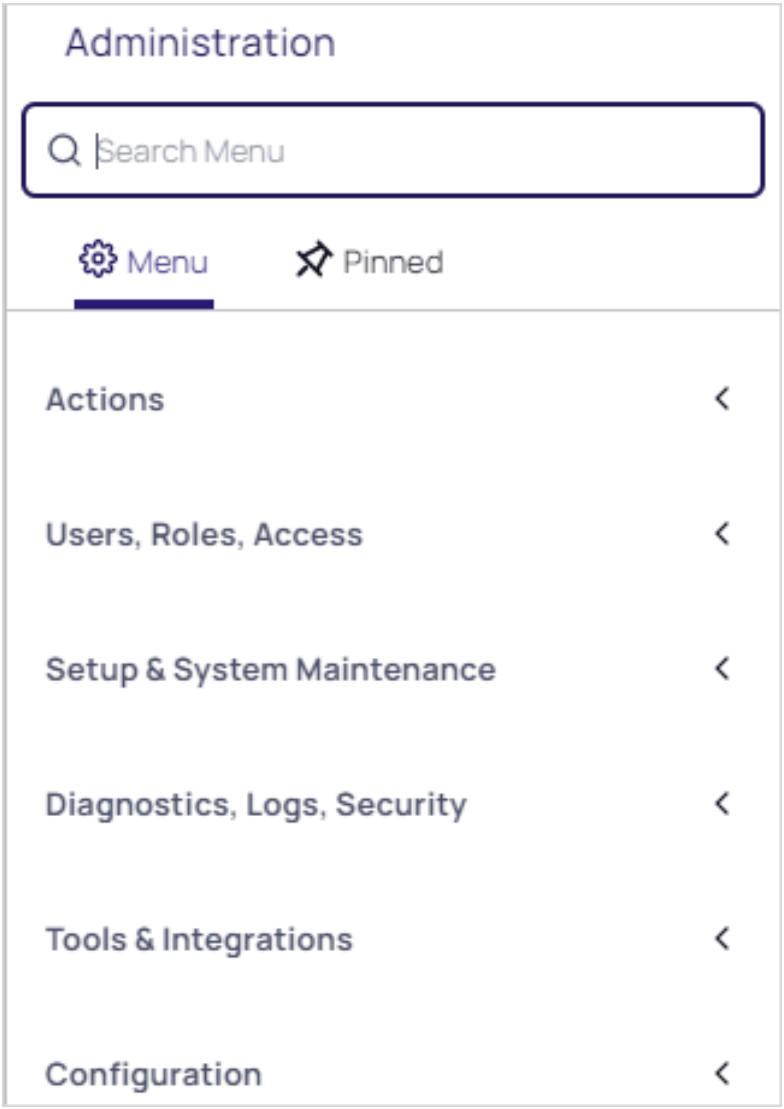
1. Create a new role called *Team Limited User*.
2. Assign all permissions of the standard user role except *Unrestricted by Teams*.

3. Assign users you want restricted by teams to this role.
4. Remove the User role from their account.

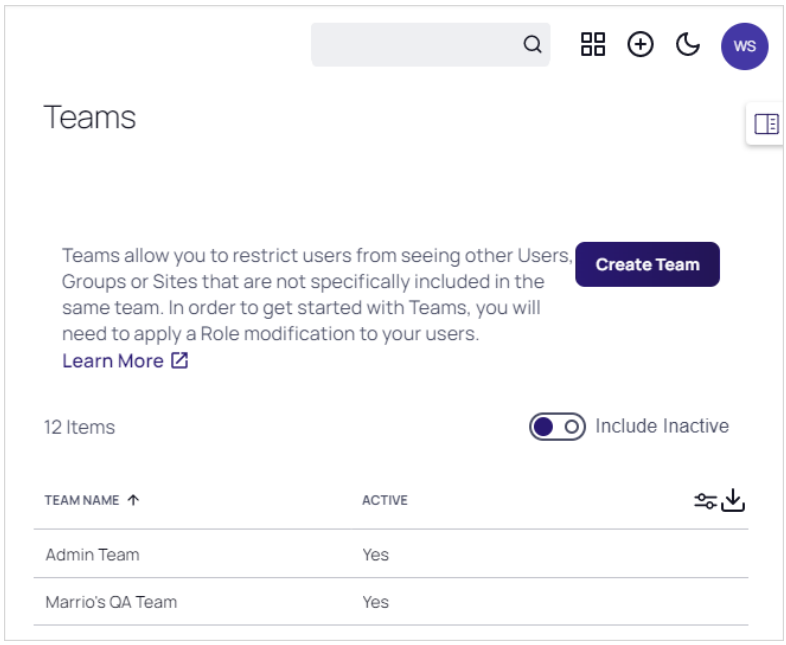
> **Note:** If you want all new users restricted by team, you can configure Secret Server to assign the Team Limited User role as the default upon creation of a new user.

Creating Teams

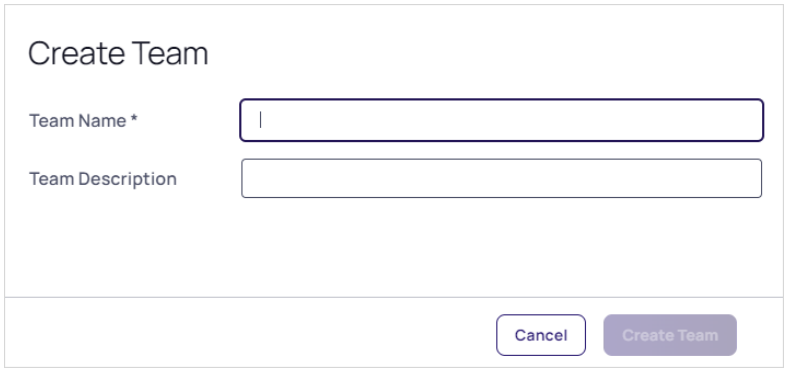
1. Navigate to **Administration**. The Admin Side Panel appears:



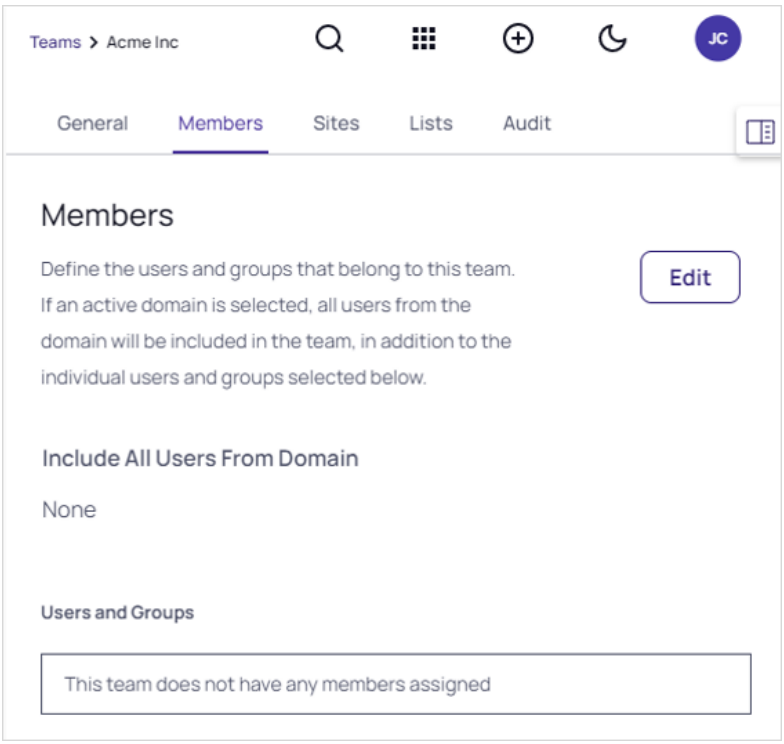
2. Click **Users, Roles, Access** and select **Teams** in the list. The Teams page appears:



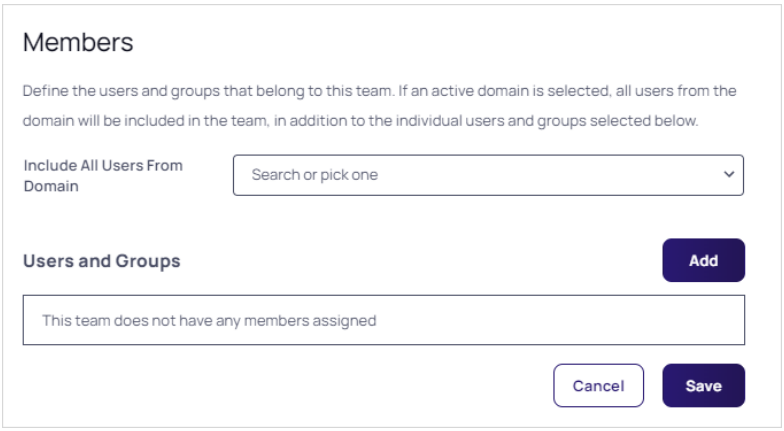
3. Click the **Create Team** button. The Create Team popup page appears:



4. Type the name for the new team in the **Team Name** text box.
5. (Optional) Type a description in the **Team Description** text box.
6. Click the **Create Team** button. The new team's Members tab appears:



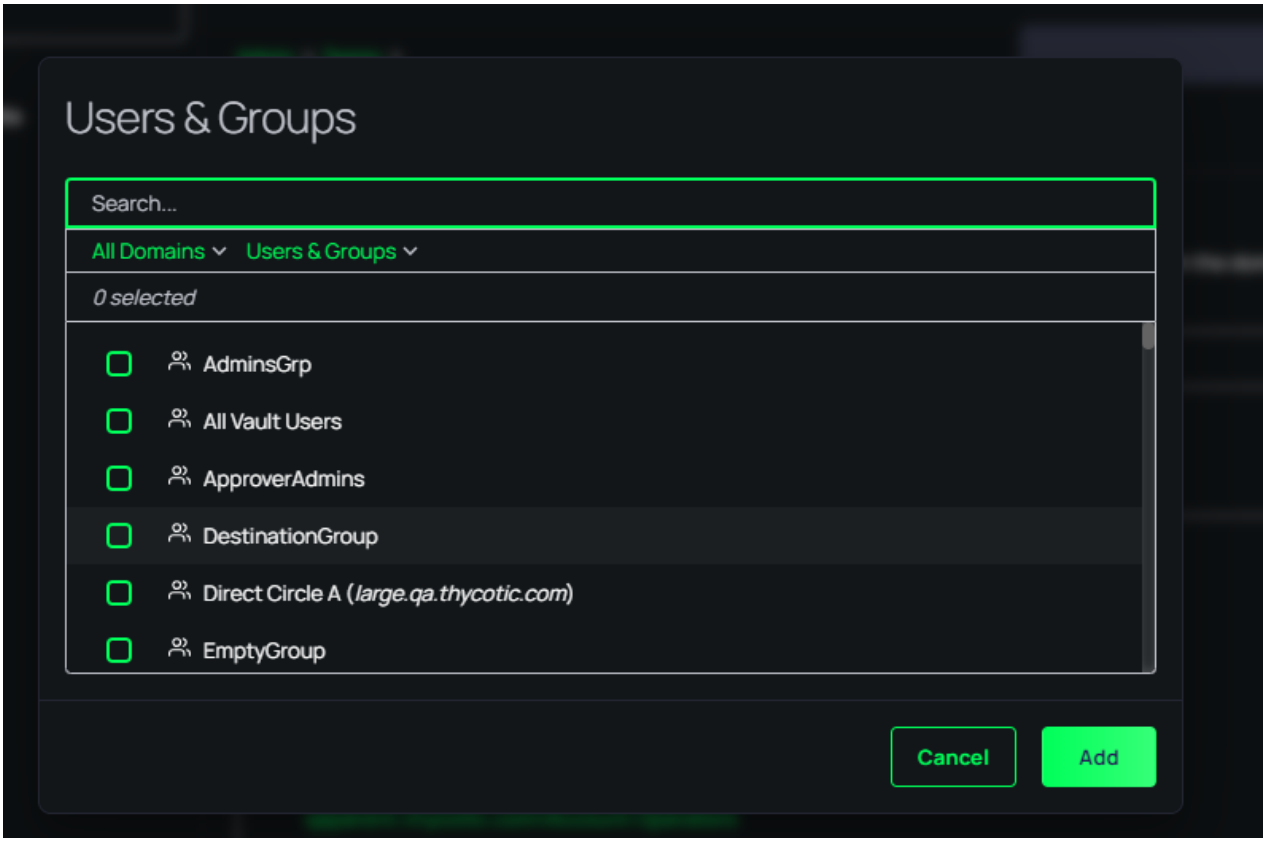
7. Click the **Edit** button. An Add Groups / Users section appears:



8. **Either** to include all users in a domain, click the first **Include All Users from Domain** dropdown list to select a domain. This only appears if you have domains available.

   Or to add individual users:

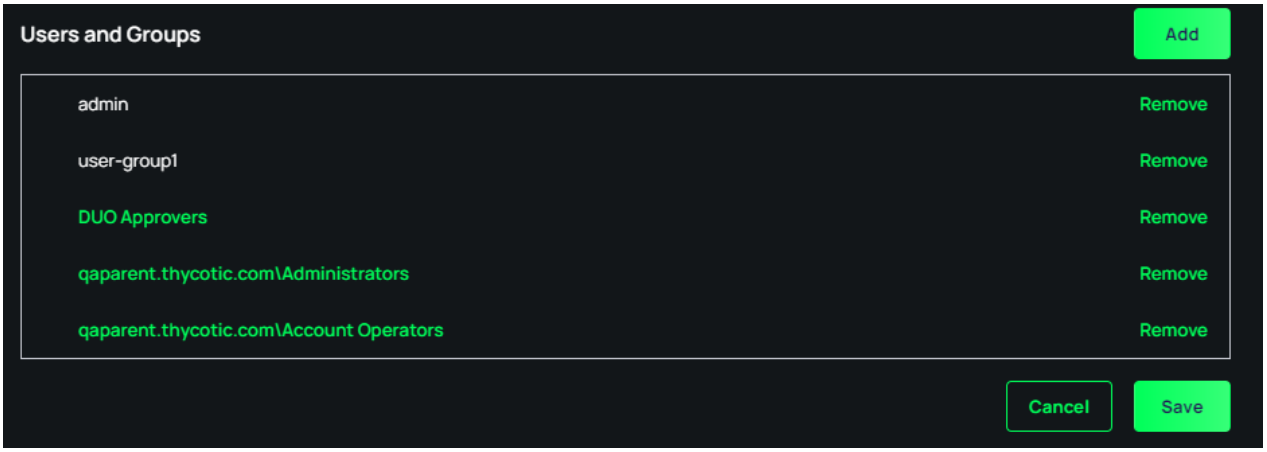
a. Click the **Add** button. The Users and Groups popup appears:



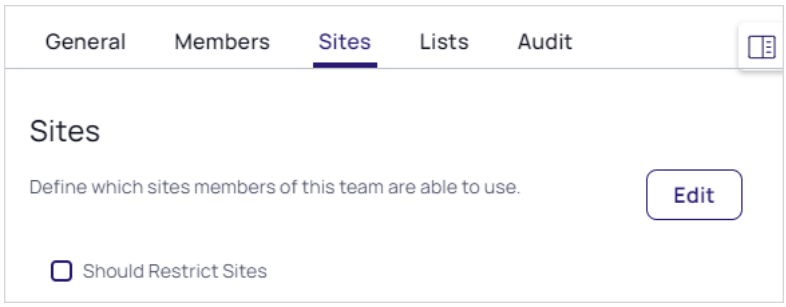
b. If needed, type the name of desired users or groups in the search box.

c. Click to select the check boxes for the desired users or groups.

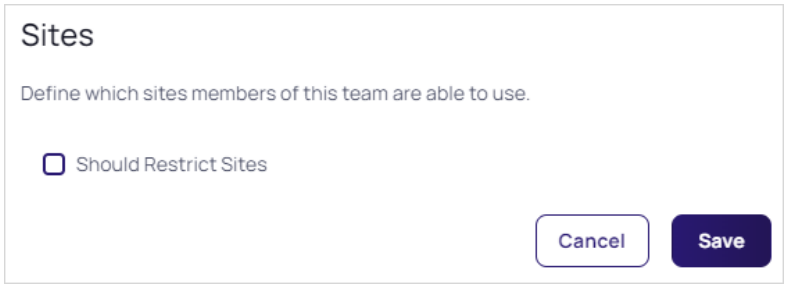
d. Click the **Add** button. The popup disappears, and your choices appear in the Users and Groups box:



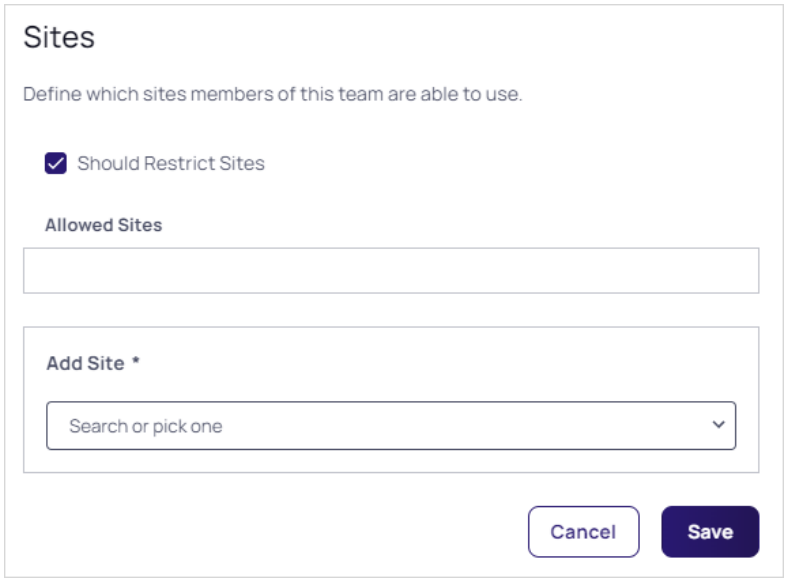
9. Click the **Save** button.
10. Click the **Sites** tab:



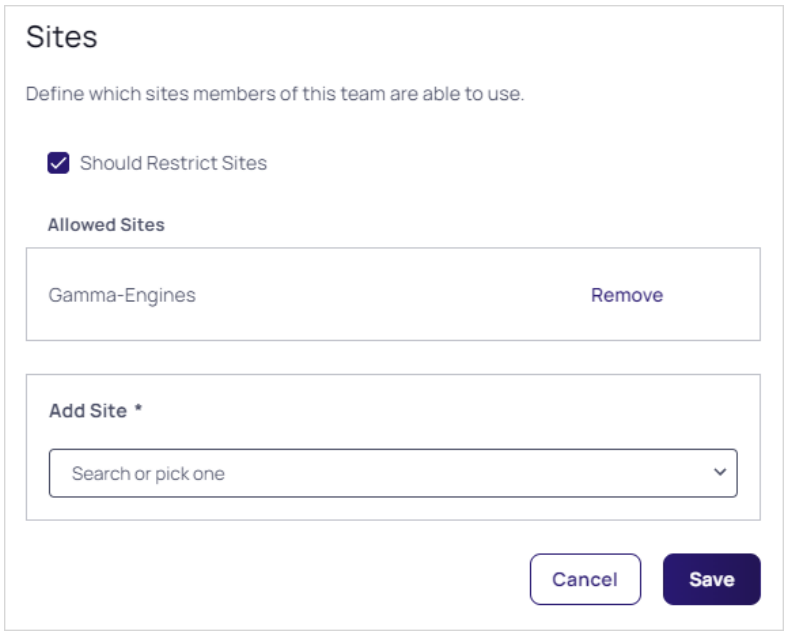
11. Click the **Edit** button. The page becomes editable:



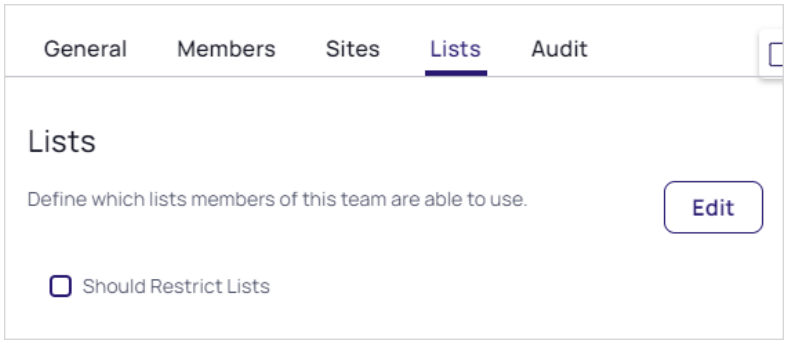
12. Click to select the **Should Restrict Sites** check box. A Site dropdown list appears:



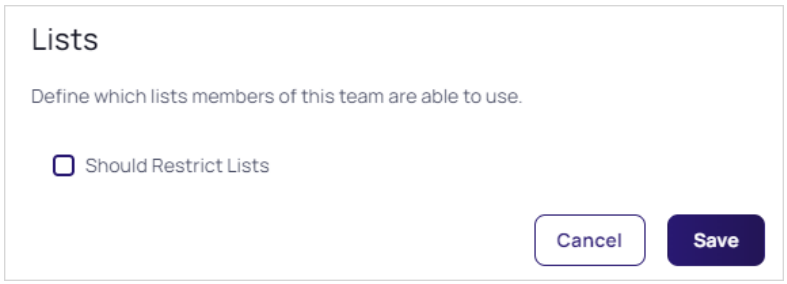
13. Click the **Add Site** list to select a site to restrict the team to. The selected site appears in the Allowed Sites box:



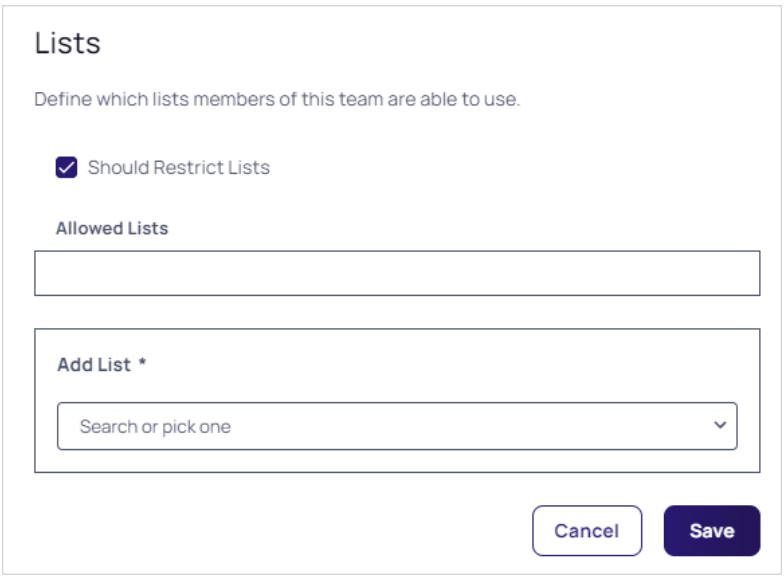
14. Click the **Save** button.
15. Click the **Lists** tab:



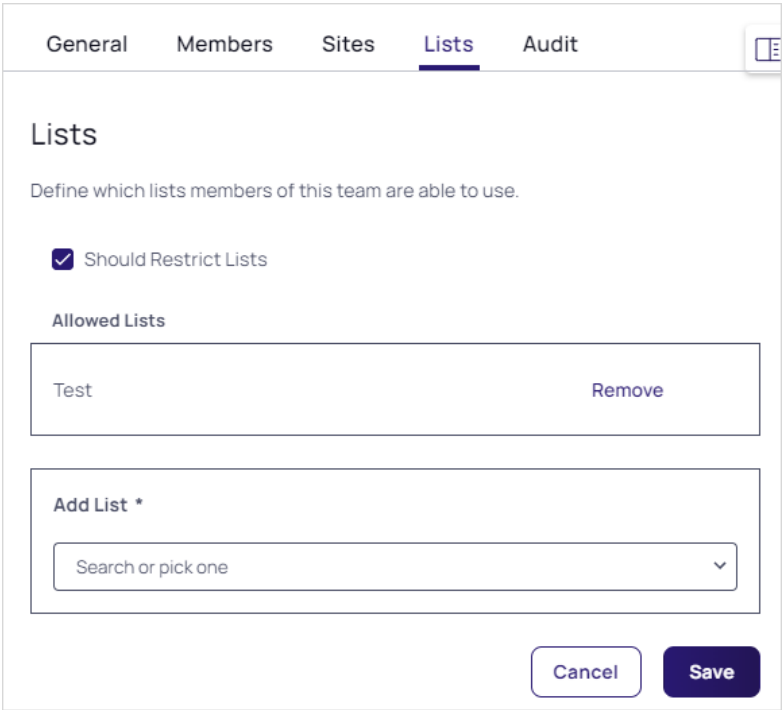
16. Click the **Edit** link. The page becomes editable:



17. Click the **Should Restrict Lists** check box.



18. Click the **Add List** dropdown list to select what lists the team members have access to. The chosen list appears in the Allowed Lists box:

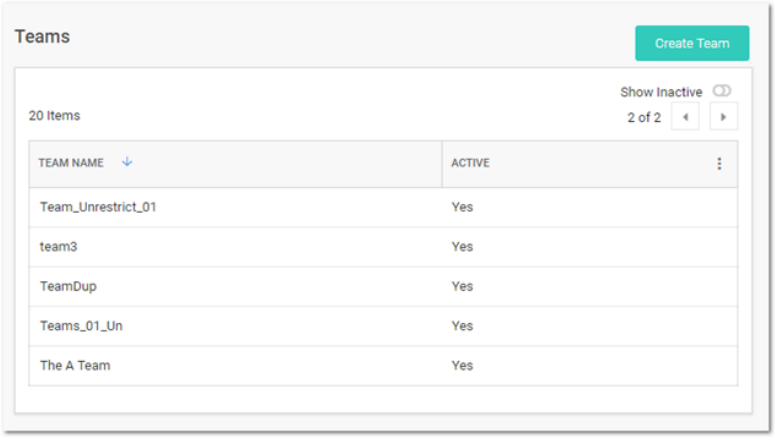


19. Repeat the process for any additional lists.
20. Click the **Save** button.

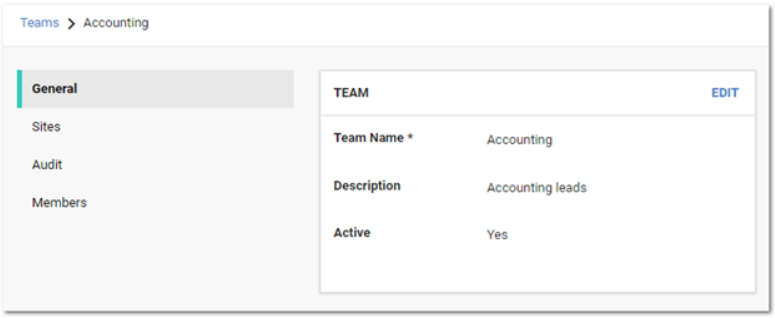
Deactivating Teams

You cannot delete teams because of auditing restrictions.

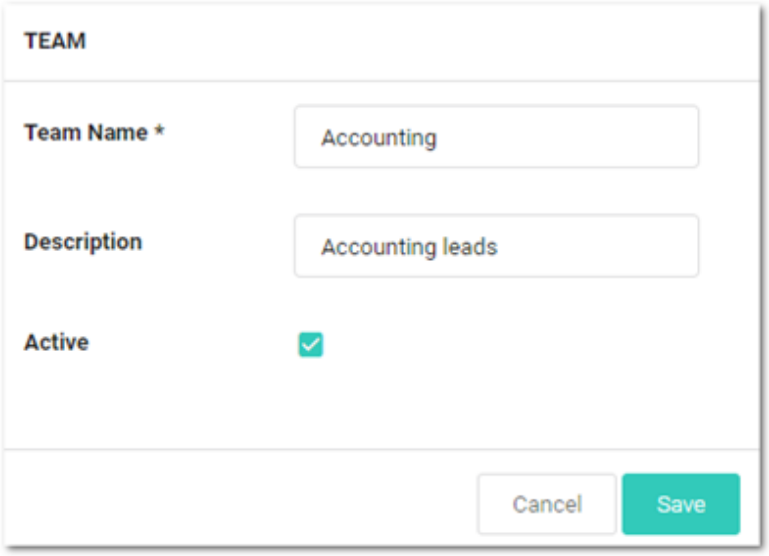
1. In Secret Server, click the **Admin** menu item. The Administration page appears.
2. Click the **Teams** button in the list. The Teams page appears:



3. Click the table row for the desired team. That team's page appears:



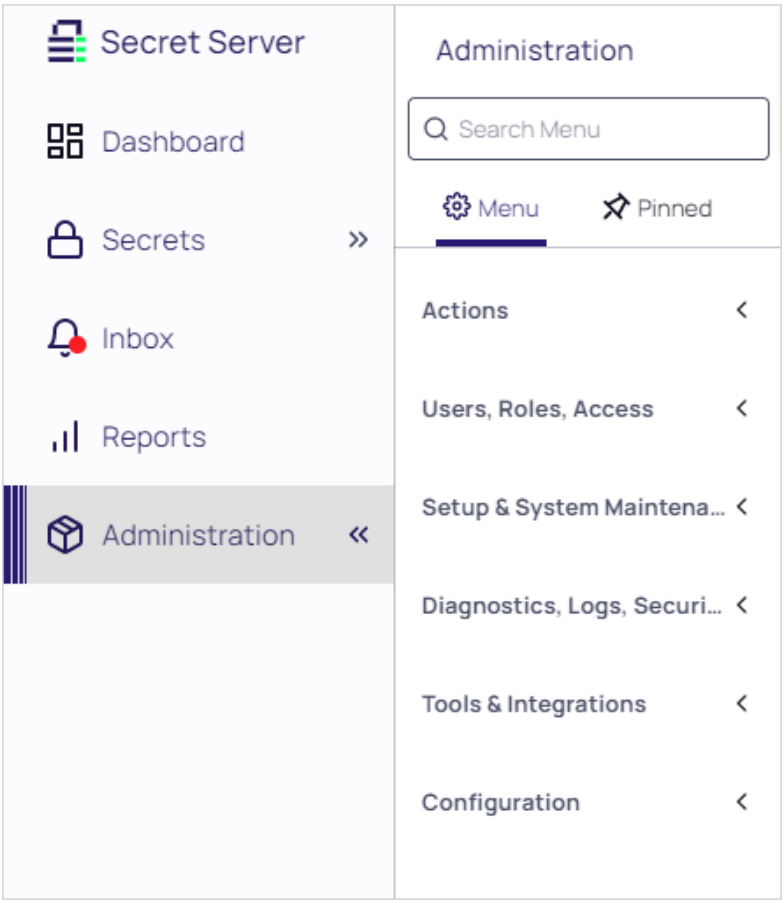
4. On the **General** page, click the **Edit** button. The tab becomes editable:



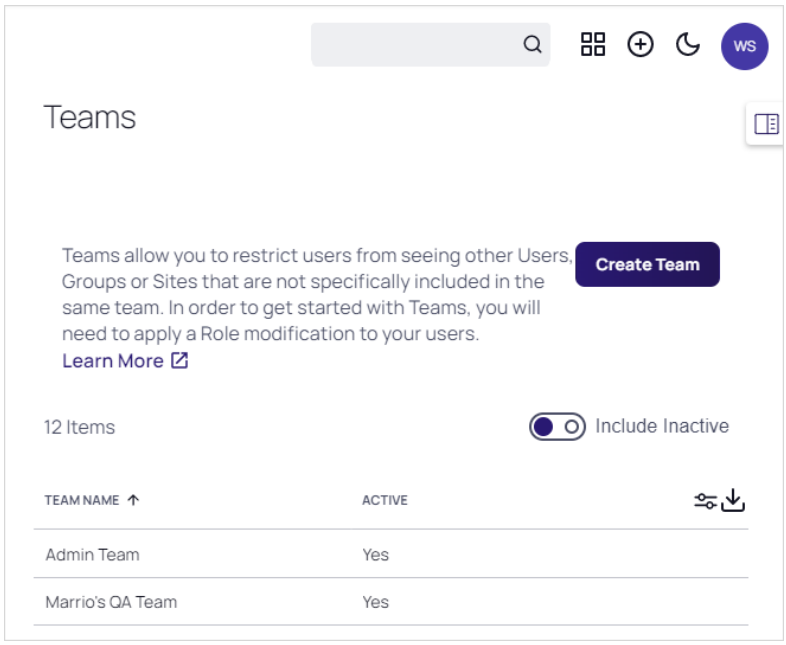
5. Click the **Active** check box to deselect it.
6. Click the **Save** button. The team is deactivated.

Editing Teams

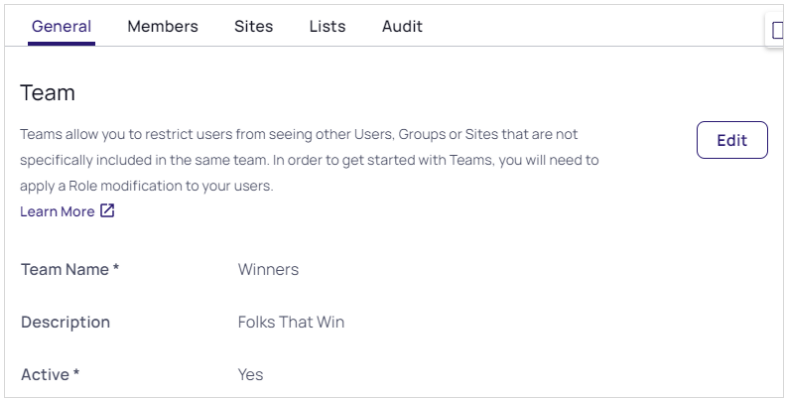
1. Navigate to **Administration**. The Admin Side Panel appears:



2. Click **Users, Roles, Access** and select **Teams** in the list. The Teams page appears:

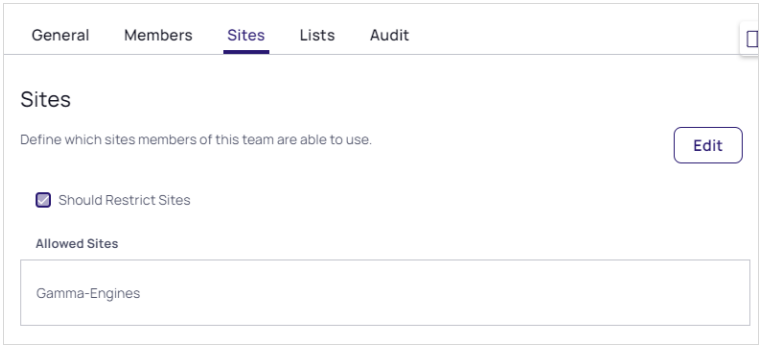


3. Click the table row for the desired team. That team's page appears on the General tab:

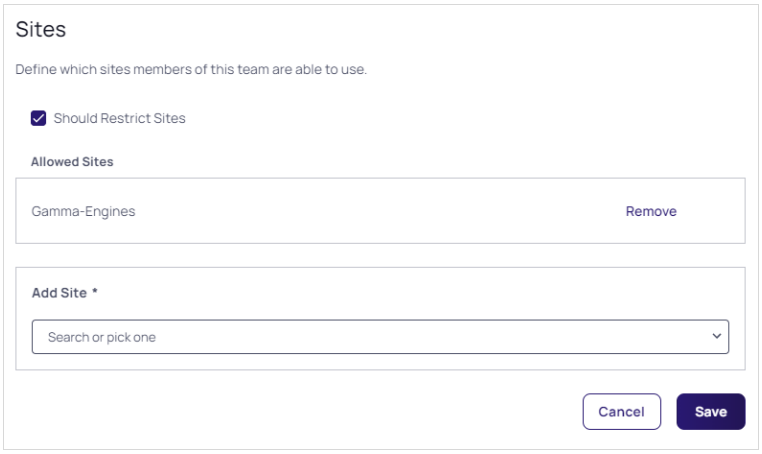


4. Click the **Edit** button to change:
   - The team name
   - The team's description
   - The team's status
5. To restrict the visible sites:

a. Click the **Sites** tab:



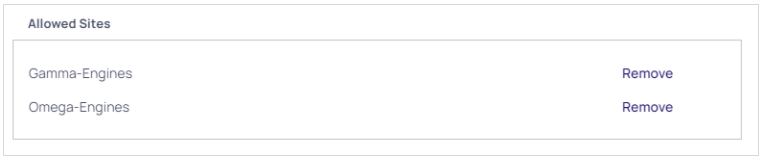
b. Click the **Edit** button. The page becomes editable:



c. If necessary, click to select the **Should Restrict Sites** check box.

d. Click the **Remove** button next to any sites you want to remove.

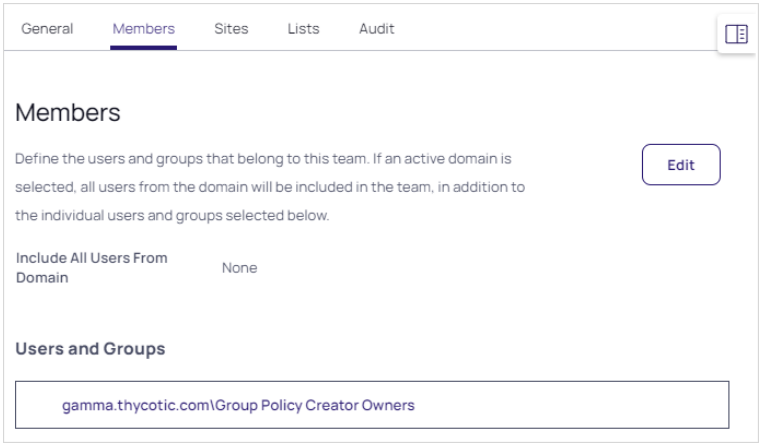
e. Click the **Add Site** dropdown list to select sites you desire to add. The selected site appears in the Allowed Sites table:



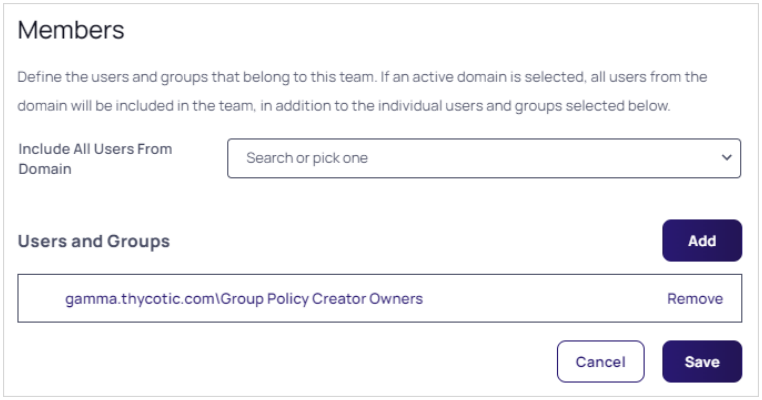
f. Click the **Save** button.

6. To edit the team's member users or groups:

a. Click the **Members** tab:



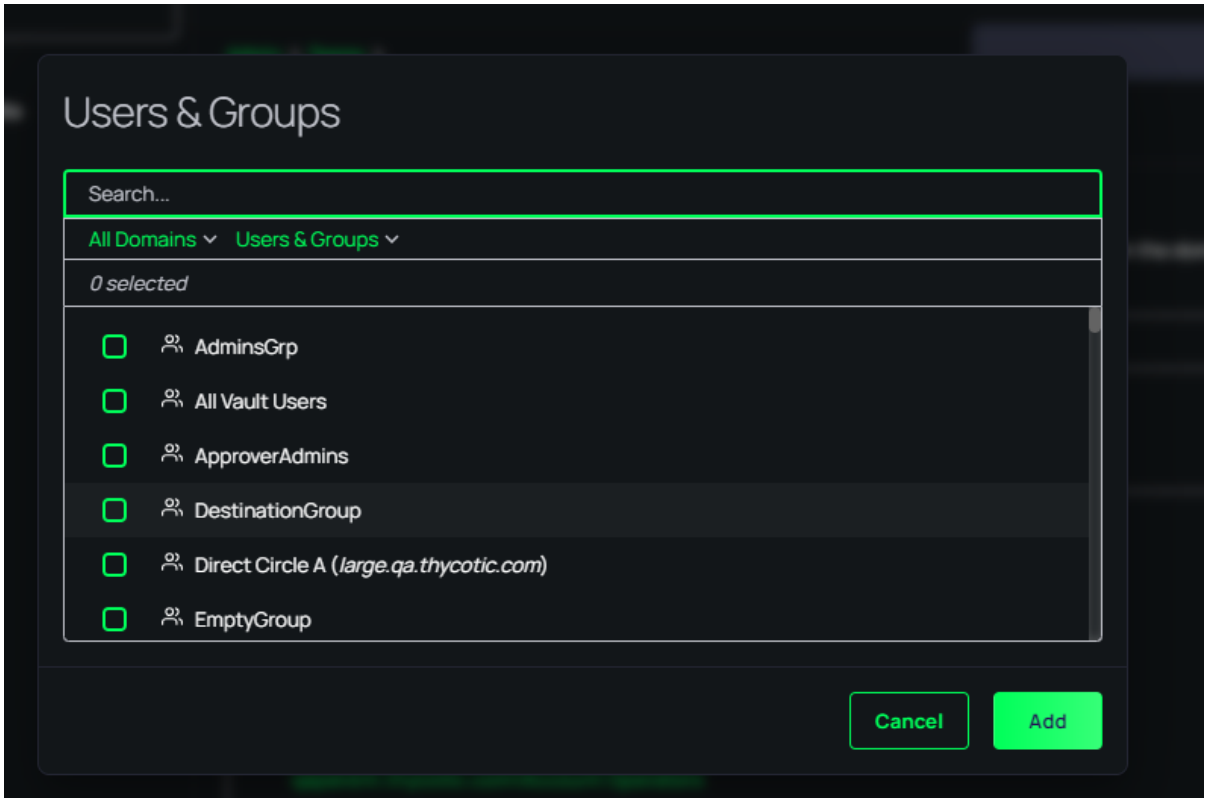
b. Click the **Edit** button. The page becomes editable:



c. **Either** to include all users in a domain, click the first **Include All Users from Domain** dropdown list to select a domain. This only appears if you have domains available.

   Or to add individual users:

i. Click the **Add** button. The Users and Groups popup appears:

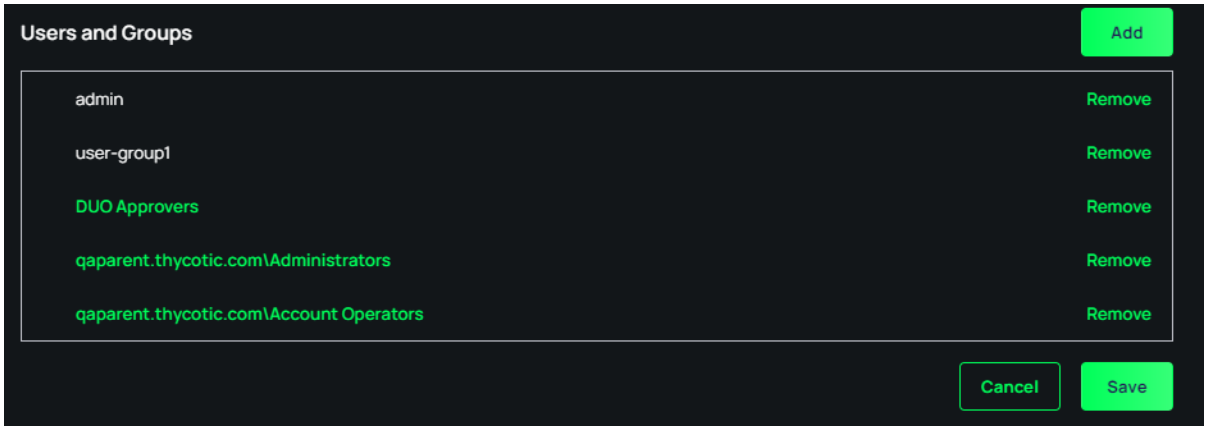


ii. Click the unlabeled domains dropdown list to select the desired domain.

iii. If needed, type the name of desired users or groups in the search box.

iv. Click to select the check boxes for the desired users or groups.

v. Click the **Add** button. The popup disappears, and your choices appear in the Users and Groups box:



vi. Click the **Save** button.

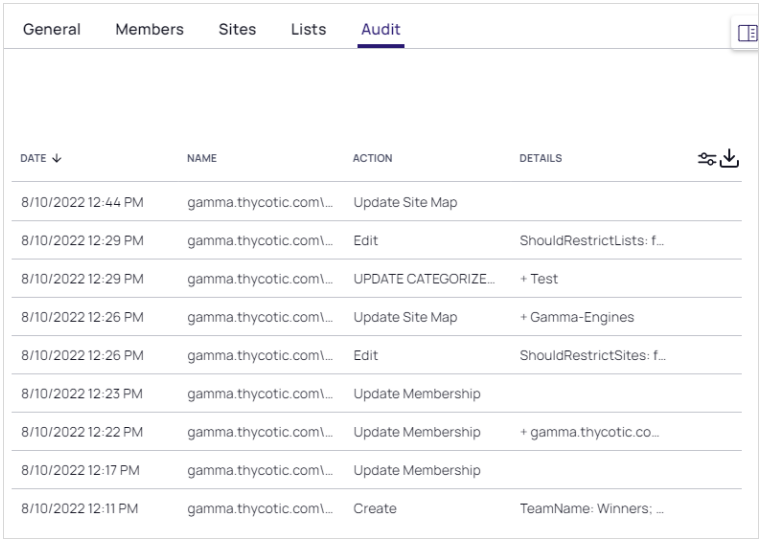
vii. Repeat the process for additional users and groups to add.

viii. Click the **Remove** button next to a user or group to delete it.

ix. Click the **Save** button.

7. View events for the team using its audit trail:

   a. Click the **Audit** tab:

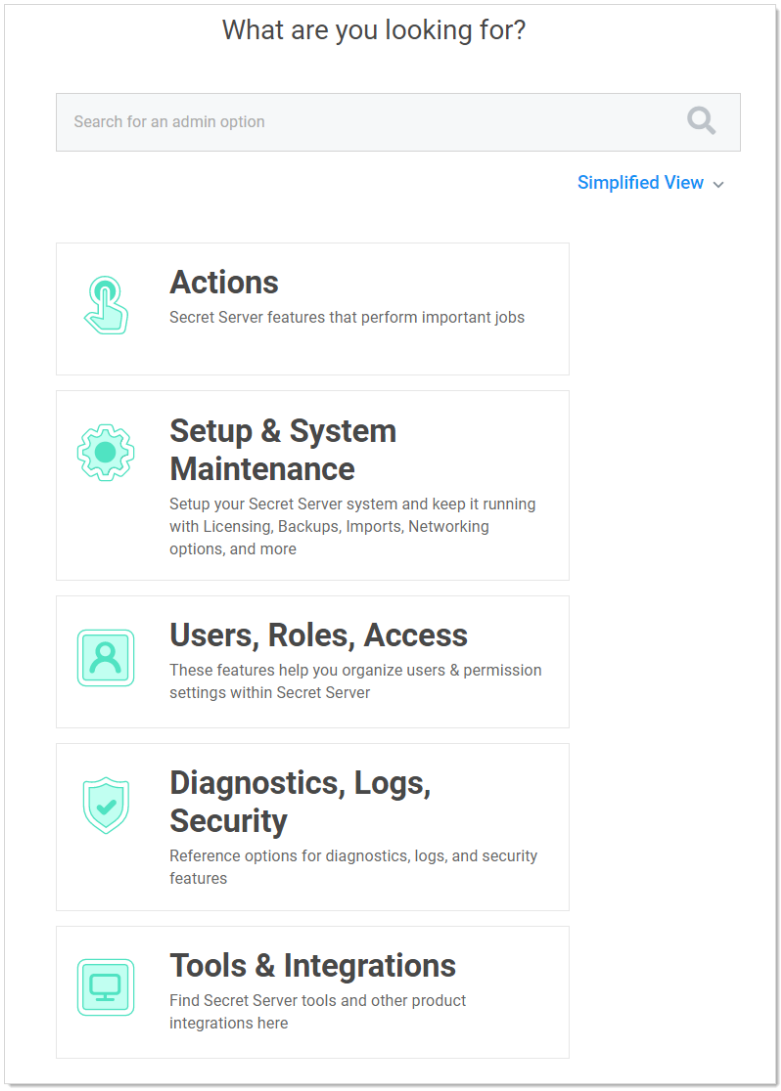


   b. Audit events occur when:

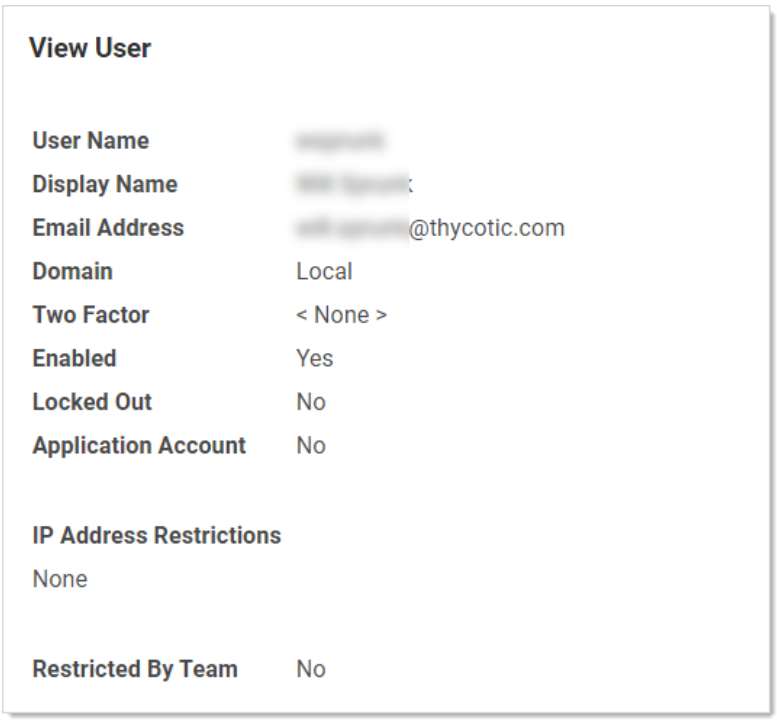
      - The team is created
      - General tab: name, description, or active status is changed
      - Sites tab: restrictions are added, removed, or changed
      - Members: users or groups are added or removed

Viewing a User's Teams

1. Navigate to **Admin > See All**. The Administration page appears:



2. Type and then click **Users** in the search text box. The View User page appears:



You can see if the user belongs to a team, and if so, what teams the user belongs to. If the Restricted by Team line says *No*, it means the user has been granted the Unrestricted by Teams permission, which means the user can view all users, groups, and sites.

Troubleshooting Teams

Users can view other users not in their teams if that user already had a connection, such as a shared secret, with the other user prior to setting up the team restrictions.

The API does not restrict who can be assigned if they use the known group ID of a user or group not in their team. This is designed so secret permissions can be saved across teams without removing the permissions of another team.

Security Compliance Standards

The Federal Information Processing Standard 140-1 (FIPS 140-1) and its successor (FIPS 140-2) are United States Government standards that provide a benchmark for implementing cryptographic software. has been tested within environments that are FIPS compliant. FIPS 140-2 compliance is built-in to Secret Server Cloud and is always on. For instructions to enabling FIPS in Secret Server On-Premises, see "Enabling FIPS Compliance in Secret Server On-Premises" on the next page.

PCI Datacenter Compliance

Secret Server can make it easier to comply with PCI-DSS requirements:

Advanced Encryption Standard

The AES encryption algorithm provides a high security level for sensitive data. The National Institute of Standards and Technology (NIST) and National Security Agency (NSA) search for a replacement for the Data Encryption Standard (DES), which had numerous issues, namely small key size and efficiency, and finally settled on AES.

> Encryption algorithms use keys to obfuscate the data. While DES only had a key size of 56 bits, AES can have a key size of 128, 192 or 256 bits. Larger keys provide more security as their size makes brute force attacks infeasible.

> To address concerns from the cryptographic community, NIST embarked on a transparent selection process. During the selection process NIST solicited designs from the global cryptographic community and voted for a winner from within fifteen finalists. The eventual winner was a team of Belgian cryptographers with their submission of the Rijndael encryption method, which became AES. For more information about the technical specifications of AES, please see the official standard.

Enabling FIPS Compliance in Secret Server On-Premises

The Federal Information Processing Standard 140-1 (FIPS 140-1) and its successor FIPS 140-2 are United States Government standards that provide a benchmark for implementing cryptographic software. Secret Server was tested and operates correctly in FIPS-compliant environments. FIPS 140-2 compliance is built-in to Secret Server Cloud and is always on.

> The Microsoft .NET implementations of AES and SHA are not FIPS certified so Secret Server uses the Windows API versions for encryption functionality which *are* FIPS certified.

See FIPS 140-2 Validation for the FIPS certificate numbers for the Windows operating systems, including the algorithm implementations that we use. Supported operating systems include Windows Server 2008 R2 and above.

Site-Specific FIPS Configuration

Individual sites are configurable for FIPS compatibility. The setting is available on the , in the Engine Default Settings dialog box. All engines on a site will use this setting, overriding the global setting, which is configured at **Administration > Configuration > Security**.

Procedure

To enable FIPS compliance in Secret Server On-Premises:

Task 1: Enable FIPS in Secret Server On-Premises

> Secret Server is unavailable and may give errors (such as "Parser Error Message: This implementation is not part of the Windows Platform FIPS validated cryptographic algorithms") until all the steps are completed.

> During Secret Server installation, if FIPS compliance for Windows has already been enabled 'InvalidOperationException' error messages may result. To resolve the issue, please contact support for assistance.

> If FIPS is enabled as part of a domain group policy, it must be disabled before the option can be enabled in Secret Server, otherwise an error may occur. It can be re-enabled using group policy once the feature has been enabled in the application.

1. In Secret Server, go to **Admin > FIPS**.
2. Click **Edit**, then click to enable the **Enable FIPS Compliance** check box.
3. Click **Save**.

Task 2: Enable FIPS in Windows

1. At the Windows command prompt, run `secpol.msc`. The Local Security Policy application appears.
2. In the left pane, drill down to **Security Settings > Local Policies > Security Options**.
3. In the right pane double-click the **System Cryptography: Use FIPS Compliant algorithms for encryption, hashing, and signing** policy. Its properties appear.
4. Click to enable the **Enabled** selection button on the **Local Security Setting** tab.
5. Click the **OK** button.
6. Close the **Local Security Policy** application.

Task 3: Reset the IIS Server

Run `iisreset` from the Windows command prompt. IIS resets.

> When using FIPS compliance mode in Secret Server, we use the NIST-certified encryption algorithms within the Windows Operating System.

Troubleshooting

If you have an endpoint that requires FIPS compliance, and Secret Server On-Premises is not configured for FIPS, you cannot establish a connection with that endpoint.

FIPS enforcement is dictated by the target machine. If the target requires FIPS compliance but the initiating system is unaware of this requirement, the initiating system might attempt to secure the connection using non-FIPS algorithms. Such attempts are rejected by the target endpoint.

Conversely, if Secret Server On-Premises is configured for FIPS compliance, it will successfully connect to target systems that do not require FIPS. The only exception to this is if the target endpoint specifically rejects a FIPS-approved algorithm that the Secret Server engine is trying to use.

Enabling FIPS compliance in Secret Server On-Premises restricts the cryptographic algorithms to FIPS-approved options during the initial handshake when connecting to the target. If the target does not support these FIPS-approved algorithms, the handshake fails, preventing a successful connection.

Thus, when troubleshooting FIPS-related connectivity, you might want to identify the algorithms supported by the target and ensure that the SSH algorithms configured in Secret Server match at least one of the target's supported algorithms.

Related Information

- NIST Cryptographic Module Validation Program Information
- FIPS information for Windows

Supported Versions of Secret Server Ancillary Tools

Below list is an overview of the supported versions for various ancillary tools and components that integrate with Secret Server. Keeping these tools updated ensures compatibility and optimal performance.

Database Support

Microsoft SQL Server

- Supported Versions: SQL Server 2014 Express, 2016, 2017, 2019, 2022.
- Notes: SQL Server 2014 Express is not recommended for production environments due to performance limitations.

See Installing and Configuring SQL Server for more details.

Web Browsers

Supported Browsers

- Chrome: Version 10.8 and later
- Edge: Version 10.8 and later
- Firefox: Version 10.8 and later
- Safari: Version 10.8 and later
- Opera: Version 10.8 and later
- Internet Explorer: Not supported

Operating Systems

Windows Server

- Supported Versions: Windows Server 2016, 2019, 2022.

See System Requirements for Secret Server for more details.

Protocol Handlers

Protocol Handler

- Protocol Handler Version 6.0.0.23 or higher is supported for managing multiple Secret Server instances. This version allows disabling the protocol handler auto-update function for forward and backward compatibility. See Managing Multiple Secret Server Instances with Protocol Handlers and Launchers for details.
- Protocol Handler Version 6.0.3.27 is the latest version as of the Secret Server 11.6.000025 release. This version includes changes to core internal functionality and requires manual redeployment or installation to end-user machines. Secret Server 11.6.000025 GA Release Notes for details.

Remote Desktop Services (RDS)

RDS Server

- Supported Windows Server Versions: 2012, 2016.
- Notes: Ensure compliance with Microsoft licensing requirements.

Scripting and APIs

PowerShell

- Supported Versions: PowerShell 5.1 and later.
- Ensure compatibility with REST API and scripting features.

REST API

- Supported Protocols: TLS 1.2 and later.
- Ensure web services are enabled for API access.

Additional Tools

RabbitMQ

- Supported Versions: Latest versions recommended.
- Use RabbitMQ Helper for installation.

.NET Framework

- Supported Versions: .NET 4.8 and later.

Best Practices

- Regular Updates: Keep all ancillary tools updated to the latest supported versions to ensure security and compatibility.
- Testing: Test updates in a staging environment before deploying to production.

- Documentation: Refer to the official documentation for each tool for detailed installation and configuration instructions.

Secret Server Security Model

> This topic only applies to **Secret Server On-Premises**.

Delinea's Secret Server is a password vault that holds the keys to the kingdom. It is therefore critical that you understand the security model of Secret Server.

What the Security Model Covers

What Is Covered

The following are part of Secret Server's security model:

- Access to data must be authenticated and authorized.
- Access to or modification of data must have auditing for accountability.
- Data is confidential and secured at rest.
- Data is confidential and secured in transit.
- The system can be highly available.

What Is Not Covered

The following are not a part of Secret Server's security model:

- Protecting against the leakage of the existence of secret data. An attacker that can read from storage will be able to tell that secret data exists, even if it is encrypted at rest.
- Protecting against memory dumps. For example, an attacker with privileged access on the web server of an distributed engine can inspect the memory of the application and potentially compromise the confidentiality of the protected data.
- Protecting against control of the web server or database server. An attacker who can manipulate the web server or database can undermine the system's security. For example, a database administrator could delete all data, and Secret Server would not be able to recover unless restored from a backup.

Confidentiality

Data is secured in transit and at rest. You should secure communications between a client and the application using Transport Layer Security (TLS). TLS sets up a secure channel for communication between the client and application and protects data in transit and against eavesdroppers. Access to the Secret Server's web interface and APIs should both be secured with TLS. Our database storage at rest secures sensitive data with AES encryption in CBC mode. Communication with the database storage can be further secured by using Microsoft's SQL Server's TLS capabilities.

> Transport Layer Security (TLS) is a cryptographic protocol used to provide secure communication between web browsers and servers, ensuring the integrity and confidentiality of data exchanged over the internet. TLS is the successor to Secure Sockets Layer (SSL) and is widely used to secure online transactions, communication, and data transfer, protecting against eavesdropping, tampering, and man-in-the-middle attacks. TLS is typically used in conjunction with HTTPS (Hypertext Transfer Protocol Secure) to provide a secure and trusted connection between clients and servers.

Availability

Secret Server can operate within a highly available configuration. You can deploy multiple web server nodes such that traffic is load balanced across them. If a web server fails, then the system relies on nodes that are still available. Likewise, you can deploy multiple distributed engines for any given site to distribute workload.

Accountability

Access to and modification of data is audited, enforcing accountability for a user attempting to access or modify sensitive data.

Authentication

Access to the web interface and APIs require authentication. Users must provide their login credentials and, if applicable, their two-factor authentication, to access the system. Without authentication, the client is unable to access any data. Authentication should be protected against eavesdroppers using TLS.

Authorization

Secret Server uses authorization to protect sensitive data that is shared in the vault. It specifically allows for RBAC (Role-Based Access Controls) that provides fine grained control over the data and controls that users have access to and ability to modify. Administrators must review the permissions that they are assigning to their users to protect against internal threats from users who have unnecessary permissions in the system. You should always use a least privilege model to assign permissions.

> Role-Based Access Control (RBAC) is a security framework that regulates access to computer resources based on a user's role within an organization, rather than by their individual identity. Users are assigned to roles, and each role is granted specific privileges and access to resources. This approach simplifies user management, improves security, and reduces the risk of unauthorized access. RBAC enables organizations to enforce least privilege access, separation of duties, and accountability, making it a widely adopted model for managing access control in enterprise environments.

Hardening Guides

> This topic only applies to **Secret Server On-Premises**.

This section discusses three hardening guides:

Overview of the Common Criteria Hardening Guide in Secret Server

The Common Criteria Hardening Guide for Secret Server provides detailed instructions for configuring Secret Server to comply with the Common Criteria (CC) for Information Technology Security Evaluation (ISO/IEC 15408). This international standard ensures that security attributes of the evaluated product are independently verified in a specific environment. Below are the key aspects covered in the guide.

> This document and all related information are legacy entries. It is the user's responsibility to test everything before making changes in a production environment.

Introduction

- Purpose: The guide is designed to help administrators configure Secret Server in compliance with Common Criteria standards.
- Audience: Intended for administrators responsible for installing, configuring, and operating enterprise infrastructure.
- Common Criteria: An international standard for security certification of computer systems, networks, and application software.

Security Hardening Checklist

- Reports: Navigate to **Reports > Security Hardening** in Secret Server to follow the checklist for securing your environment.

Configuring TLS

- TLS Requirement: Common Criteria certification requires enabling Transport Security Layer (TLS).
- Disabling TLS 1.0: TLS 1.0 must be disabled as it is no longer considered secure.
- Diffie-Hellman Hardening: Configure servers with stronger Ephemeral Diffie-Hellman settings.
- Restricting Cipher Suites: Only specific cipher suites are allowed for TLS communication.
- IIS Crypto Tool: Use the IIS Crypto tool to change cipher suites.
- HTTPS/SSL: All connections to the Secret Server web page must use HTTPS/SSL.
- TLS Auditing: Enable auditing for TLS connections and failures.
- Active Directory: Ensure TLS is configured with Active Directory.
- Syslog: Configure TLS with Syslog for secure logging.

Additional Common Criteria Configurations

- X.509v3 Certificates: Install and configure certificates following the X.509v3 standard.
- DPAPI: Enable the Windows Data Protection API (DPAPI) to protect the master encryption key file.
- FIPS Mode: Enable Federal Information Processing Standard (FIPS) mode in Secret Server On-Premises for cryptographic functionality.
- Zero Information Disclosure: Configure Secret Server to hide unnecessary information, such as detailed error messages and application version numbers.
- Login Banner: Configure a login banner to display the user policy agreement.
- Account Lockout: Implement account lockouts to prevent repeated unsuccessful login attempts.
- Disabling "Remember Me": Disable the "Remember Me" login function for security reasons.
- SQL Server Configuration: Install Microsoft SQL Server on the same machine as Secret Server and use Windows authentication mode.
- Service Account: Create a service account to run the Secret Server IIS Application Pool and configure necessary permissions.

Enabling Common Criteria Security Hardening

The PDF version of this online document is automatically generated and thus may have minor formatting anomalies.

This document is not updated with every Secret Server release–some minor releases do not affect the guide's contents and thus do not warrant a document update.

Introduction

This document is closely associated with the Security Hardening Report in Secret Server (Click **Reports > Security Hardening**) and with the "Security Hardening Guide" on page 1402, which provides information on security hardening beyond Common Criteria. We recommend having those available while reading this document.

Overview

Secret Server made several security enhancements to achieve Common Criteria (CC) certification. These features are available in all versions of Secret Server 10.4 and later. Due to their stringency and need for additional user configuration, not all these features are enabled by default by our standard installer.

This guide provides the information an administrator needs to configure Secret Server 10.4 and above in compliance with the Common Criteria evaluated configuration. Follow this guide in its entirety to ensure each parameter setting matches those evaluated and certified as secure by Common Criteria standards.

Audience

This document is for administrators who are responsible for installing, configuring, and operating enterprise infrastructure for their organization. To use this guide, you must have knowledge of your organization's network infrastructure and applicable policies. In addition, you must have administrative access to configure your operational environment.

What Is Common Criteria?

The Common Criteria for Information Technology Security Evaluation (ISO/IEC 15408), known as "Common Criteria," is an international standard for security certification of computer systems, networks, and application software. The certification ensures that claims about the security attributes of the evaluated product were independently verified in the evaluated configuration in the same specific environment. The certification assumes a specific evaluated configuration and does not validate any security claims when the product is used outside of that configuration.

Consider using the following general best practices, aligned with ISO 15408:

- Documenting and evaluate Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) to meet security profiles.
- Performing regular security audits, vulnerability assessments, and penetration tests to maintain compliance.
- Implementing robust logging and monitoring for both application and database activities.

Procedures

Security Hardening Checklist

After installing Secret Server, navigate to the **Reports > Security Hardening** tab, and follow the checklist to ensure your environment is as secure as possible:

Note: See the "Security Hardening Guide" on page 1402 for details.

Configuring TLS

To achieve Common Criteria certification on Secret Server, you must enable Transport Security Layer (TLS).

.

Manually Disabling TLS Version 1.0

TLS 1.0 is no longer considered secure, so it is important to disable this version of the protocol on Secret Server. To do this, follow the instructions in the "Manually Disabling TLS v1.0" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

TLS Diffie-Hellman Hardening Overview

For information on configuring your servers with stronger Ephemeral Diffie-Hellman hardening, see the "TLS Diffie-Hellman Hardening Overview" in the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Restricting Server Cipher Suites for TLS

Allowed Suites

Common Criteria certification requires restricting the cipher suites configured on your server to the following:

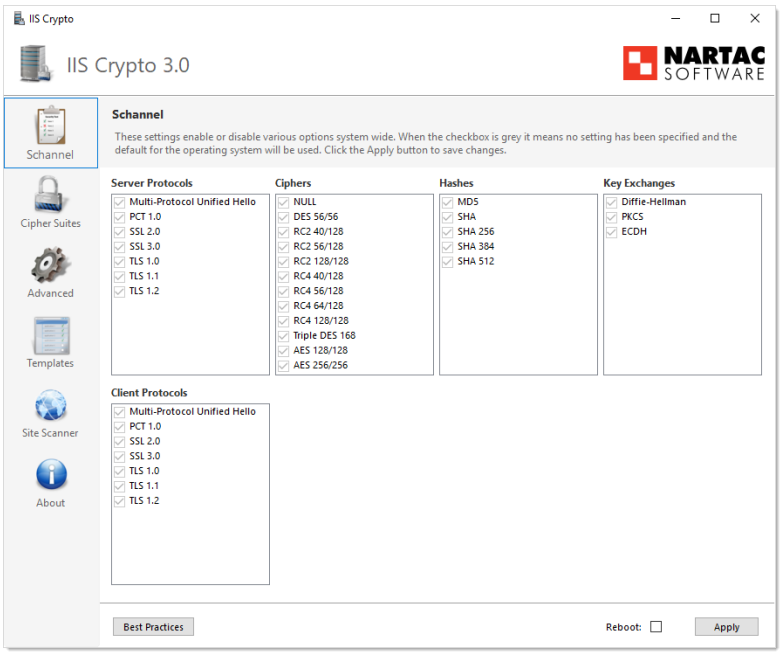
- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246.
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246.
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246.
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288.
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246.
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246.
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288.
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289.
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289.
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289.
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289.
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289.
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289.
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289.
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289.

Restricting them can cause communication issues with other servers if they are unable to communicate using any of the above ciphers. In that case, you need to modify those servers to include these cipher suites to securely communicate according to Common Criteria guidelines.

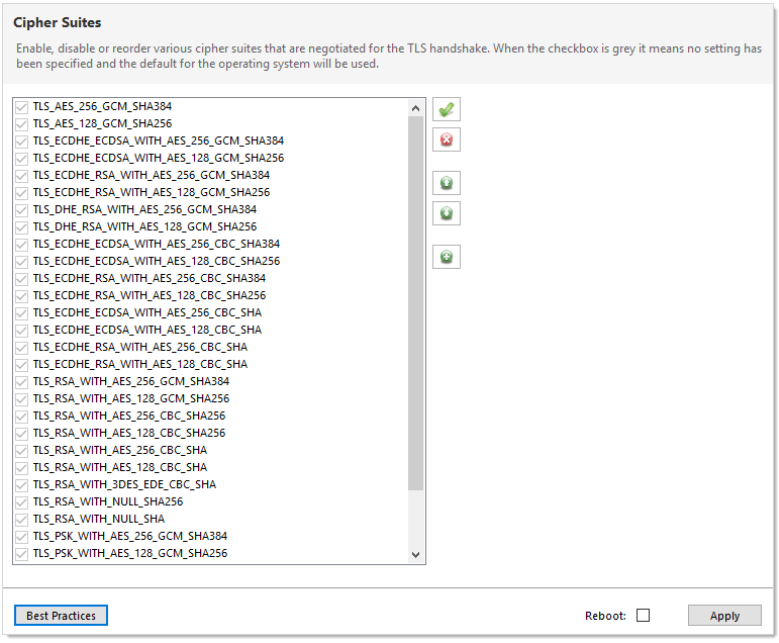
Changing Cipher Suites with the IIS Crypto Tool

One way to change the cipher suites on a computer is to use the free IIS Crypto tool:

1. Download the GUI version of the tool at: https://www.nartac.com/Products/IISCrypto/Download
2. Run the tool:



3. Click **Cipher Suites** button on the left. The Cipher Suites window appears:



4. Click the **Uncheck All** button to uncheck all cipher suites.
5. Find and click to select the suites in the list above.
6. Click the **Apply** button.

Configuring TLS with IIS

Common Criteria certification requires using HTTPS/SSL for all connections to the Secret Server Web page. To do this, follow the instructions in the "TLS Configuration with IIS" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Enabling TLS Auditing

To have Secret Server audit TLS connections and connection failures, follow the instructions in the "Configuring Auditing for TLS Connections" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Configuring TLS with Active Directory

To ensure that TLS is configured with Active Directory Follow the instructions in the "Configuring TLS with Active Directory" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Note: If you have any existing domains configured in Secret Server, you must edit them and enable LDAPS on each one.

Configuring TLS with Syslog

To configure TLS with Syslog, follow the steps in the "Configuring Syslog/CEF External Audit Server" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Additional Common Criteria Configurations

Configuring X.509v3 Certificates

See the "Configuring X.509v3 Certificates" section of the Common Criteria Hardening Guide for instructions on installing and configuring certificates on the Secret Server Web servers.

Enabling DPAPI

The Windows Data Protection API (DPAPI) is a pair of functions that allow access to operating-system-level data protection services to protect the master encryption key file, encryption.config.

To enable DPAPI, follow the instructions in the "Verify DPAPI Setting Is Enabled" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Enabling FIPS Mode

To configure your server and Secret Server to use the Federal Information Processing Standard (FIPS), follow the instructions in the "Verify FIPS Mode Is Enabled" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Note: Also see: "Enabling FIPS Compliance in Secret Server On-Premises" on page 1304

Ensuring Zero Information Disclosure

To comply with Common Criteria requirements, you must configure Secret Server to not display any unnecessary information. This applies to unhandled errors as well as the application version number.

Configuring Custom Error Messages

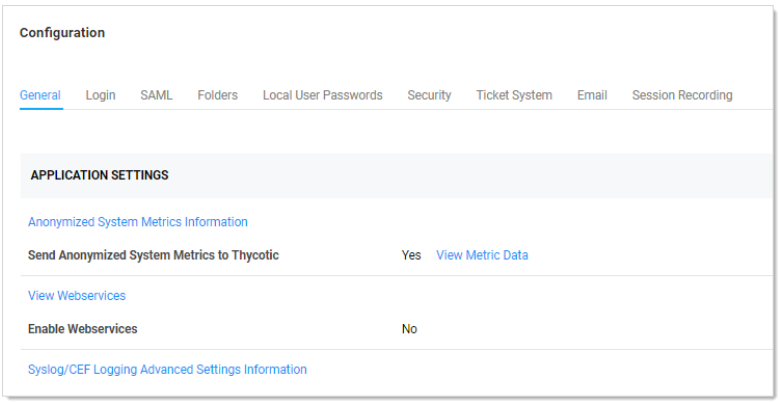
To hide detailed error messages and display a custom message when an unhandled error occurs:

1. Open `https://<your_secret_server_url>/ConfigurationAdvanced.aspx` in your browser.
2. Scroll to the bottom of the page and click the **Edit** button.
3. Type the message you want displayed to users in the **Zero Information Disclosure Message** text box.
4. Click the **Save** button.

Hiding the Application Version Number

To hide the application version number in the application header and footer:

1. Go to **Admin > Configuration > Security**.
2. Click the **Edit** button. The Configuration page appears:



3. Click the **Security** tab.
4. In the **Web Services** section of the page, click to select the **Hide Secret Server Version Numbers** check box.
5. Click the **Save** button.

> For diagnostic purposes, the application version number is still displayed on the Diagnostics page. Make sure that permissions to this page is limited to employees that may need to access this page when contacting Delinea technical support.

Configuring the Login Banner

For Common Criteria compliance, when a user first logs in, the login banner must reveal the user policy agreement and force that user to agree to the policy before logging into Secret Server. To configure the Login Banner according to Common Criteria guidelines, follow the instructions in the "Configuring the Login Banner" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Configuring Account Lockout

To access Secret Server, users must login with local or domain credentials. To comply with Common Criteria, Secret Server must use "account lockouts" to prevent repeated unsuccessful login attempts. Configurable by an Secret Server admin, an account becomes inaccessible after a defined number of unsuccessful authentication attempts until an admin unlocks the user's account.

To configure settings for account lockouts:

1. Navigate to **Admin > Configuration.**
2. Click the **Login** tab.
3. Click the **Edit** button.
4. Adjust the number in the **Maximum Login Failures** text box. The default is five attempts.

To Unlock a user's account:

1. Navigate to **Admin > Users > Select the User**.
2. Click the **Edit** button.
3. Click to deselect the **Locked Out** check box.
4. Click the **Save** button.

Disabling "Remember Me" Logins

A browser's "remember me" login function stores the user's login name and password so the user does not need to enter it again on that browser, which is both convenient and insecure. To disable "Allow Remember Me" during logins, follow the instructions in the "How to Disable Allow Remember Me during Logins" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Configuring SQL Server

To meet Common Criteria requirements, Microsoft SQL Server must be installed on the local machine—the same as the Secret Server Web server. During the install process for MS SQL, ensure that you use Windows authentication mode.

If you have an installed instance of Secret Server that does not meet this requirement, you can migrate the remote database to the server hosting Secret Server. If MS SQL Server is not installed and configured on the Secret Server, you must install it. The server must be configured with enough RAM, storage space, and processors to support running MS SQL Server and the Web site simultaneously. After copying the database, you can go to **Admin > Database** to point Secret Server to the new database location.

Because the database must be installed locally with the Secret Server Web application for Common Criteria compliance, Secret Server is not fully compliant when running multiple nodes.

ISO 15408 emphasizes securing the database against unauthorized access, misuse, or compromise, without specifying its physical or logical location. Key aspects include access control, data integrity, and confidentiality. Therefore, we recommend the following Microsoft Windows Considerations:

- Placing the database in a secure network segment (e.g., DMZ or private subnet).
- Using role-based access controls (RBAC) to restrict database access.
- Implementing encryption for data at rest and in transit to safeguard sensitive information.

Location of the Application

Applications must enforce security policies, securely communicate with the database, authenticate users, and protect sensitive operations.

- Applications can reside locally (on the same server as the database) or remotely (on a separate server), depending on performance and security needs.
- Ensure secure communication between the application and database (e.g., using TLS/SSL).
- In multi-tenant environments, isolate applications using Windows features like AppLocker or virtualization.

Application-to-Database Authentication

The authentication mechanism must establish mutual trust between the application and database while preventing unauthorized access. Therefore, we recommend the following Microsoft Windows considerations:

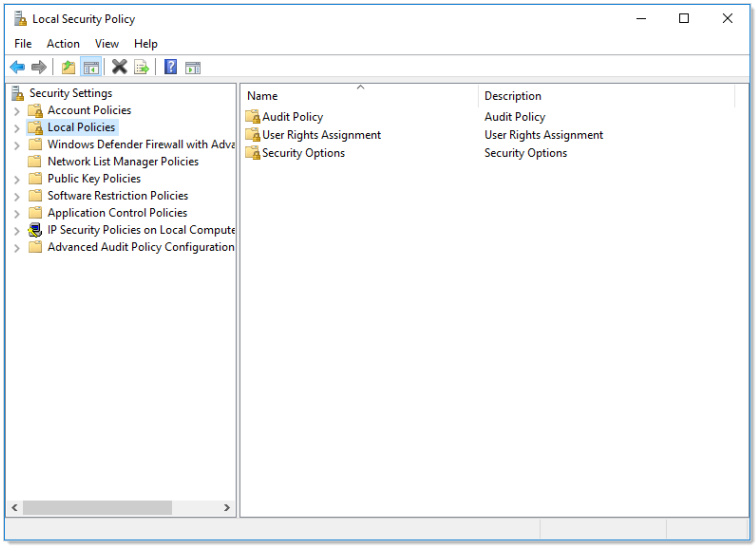
- Using Windows Authentication with Active Directory for centralized, secure credential management.
- Using Integrated Security (Kerberos or NTLM) for database connections to avoid embedding credentials in code.
- Applying the principle of least privilege, ensuring the application has only the required database permissions.
- Rotating credentials regularly and store them securely (e.g., using Windows Credential Manager or Azure Key Vault).
-

Running the IIS Application Pool with a Service Account

To use Windows authentication to access the SQL database, you should create a service account. To run the Secret Server IIS Application Pool with a service account:

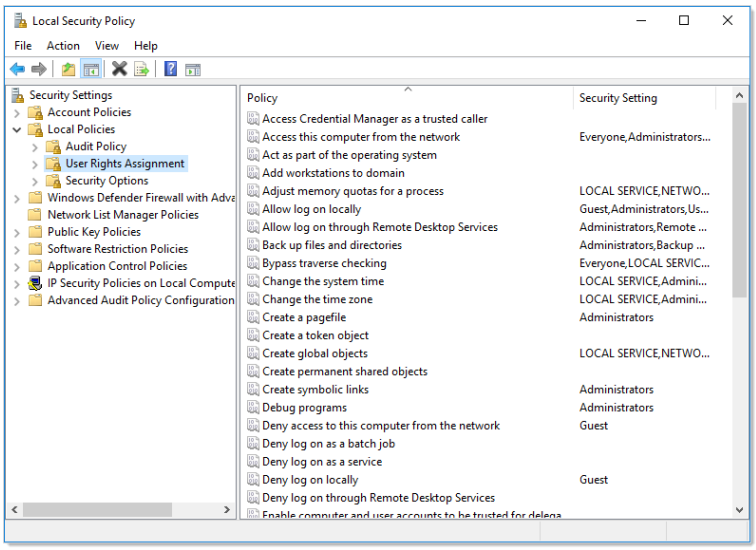
1. Open a command prompt window, change the directory to your .NET framework installation directory (usually `C:\Windows\Microsoft.NET\Framework…`) using the `cd` command.
2. Type `.\aspnet_regiis -ga <user_name>` and press **<Enter>**. The username is from the MS SQL Server user account.
3. Give your service account "modify" access to `C:\Windows\TEMP`.
4. Open the Local Security Policy App from your start menu.
5. Grant batch logon permissions to your service account:

a. Open the Local Security Policy Console (search for and open `secpol.msc`):

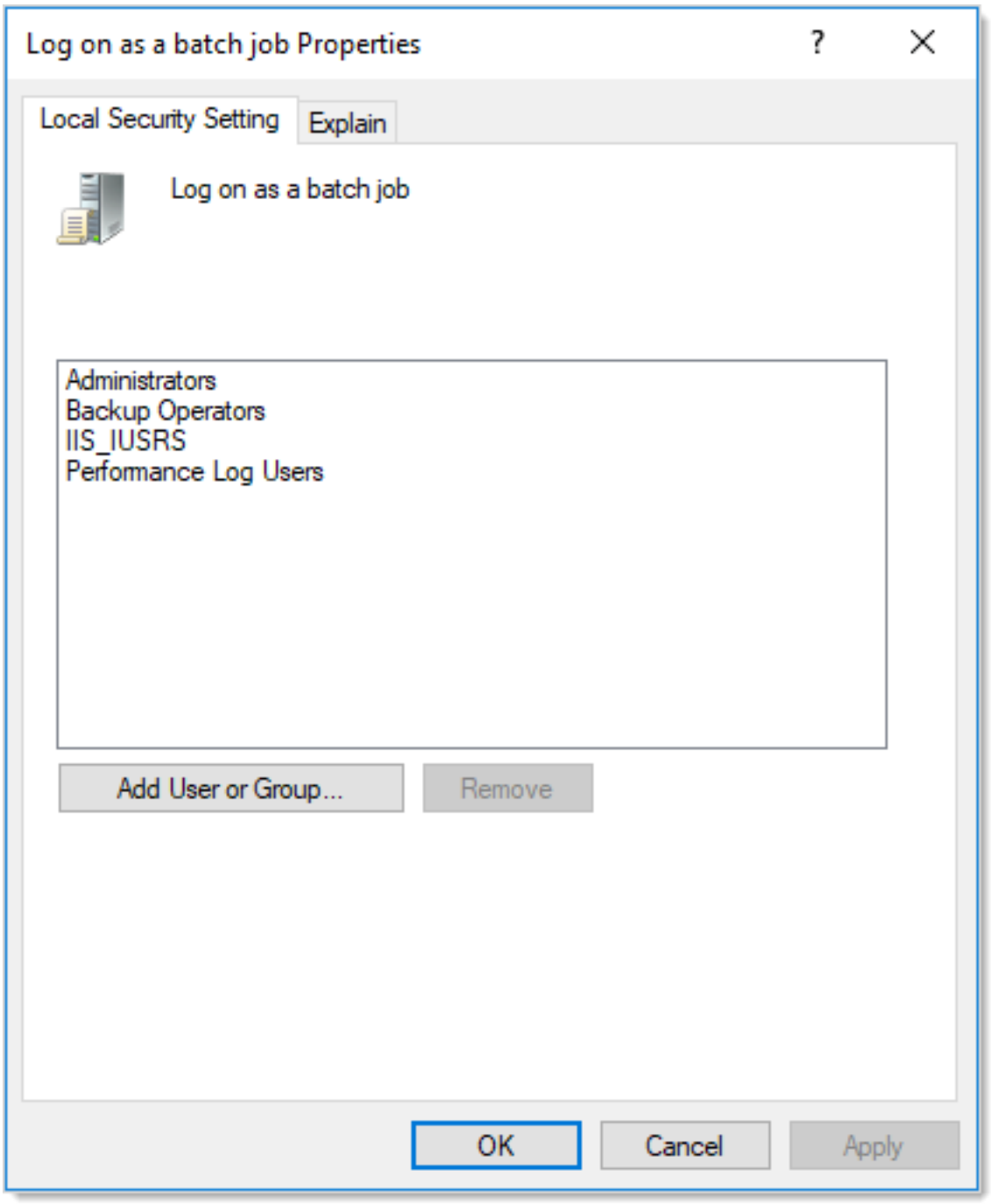


b. Expand the **Local Policies** folder (not shown).

c. Click to select the **User Rights Assignment** folder.



d. Right-click **Log on as a batch job** in the right panel and select **Properties**.



e. Click the **Add User or Group** button.

f. Add your service account.

g. Click the **OK** button.

> If you use group policy to enforce "Log on as a batch job" and have group-managed service accounts, that will overwrite any local permissions to "Log on as a batch job" on all computers that have the policy applied. Using the local security policy is a safer option if you are not sure about your usage across your domain.

5. Grant "Impersonate a client after authentication" permission to the service account under **User Rights Assignment** the same way "Log on as a batch job" was assigned above.
6. If you now get a "Service Unavailable" error after applying "Log on as a batch job" permissions:
   a. Update your group policy settings (**Start > Run > Cmd** and type `gpupdate /force`) and restart the Windows Process Activation service.

> For more information, see "Running the IIS Application Pool As a Service Account" on page 60.

Assigning Common Criteria Roles and Permissions

See the "Common Criteria Roles and Permissions" section of the Common Criteria Hardening Guide.

Managing User Passwords

See the "Managing User Passwords" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Configuring Secret Templates

To enable only the secret templates that are certified Common Criteria compliant and to set Common Criteria-compliant password policies on those templates, see the "Configuring Secret Templates" and "Configuring Password Policy for Secret Templates" sections of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Setting Authentication Strength for Non-Password Credentials

See the "Authentication Strength for Non-Password Credentials" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Configuring Remote Password Changing for SSH Key Rotation

See the "Configuring Remote Password Changing for SSH Key Rotation" section of the Common Criteria Hardening Guide.

Configuring External Auditing

Connecting to an External Audit Server

To connect to an external syslog/CEF audit server, see the "Security–Connecting to an External Audit Server" and "Configuring Syslog/CEF External Audit Server" sections of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

Configuring Local Windows Event Log Auditing

See the "Configuring Local Windows Event Log Auditing" section of the "Overview of the Common Criteria Hardening Guide in Secret Server" on page 1310.

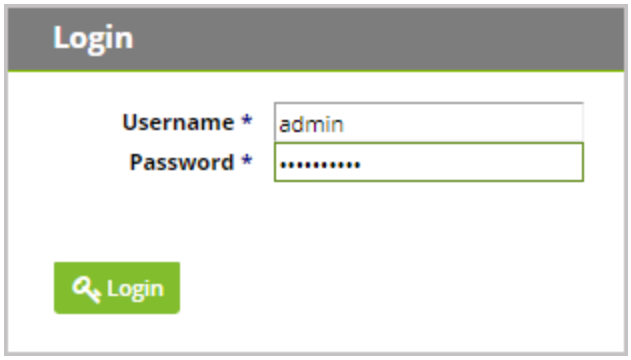
Accessing Your System

Browser Compatibility

See "Secret Server Major Browser Support " on page 85

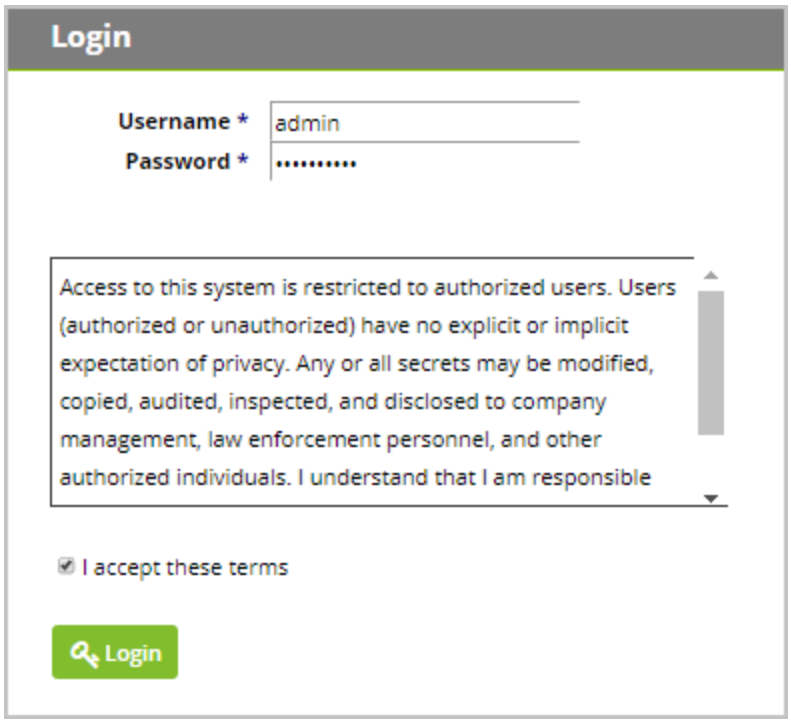
Web Interface

To access Secret Server, navigate to your organization's URL and login using your administrator credentials. The first time you login your screen will look like this:



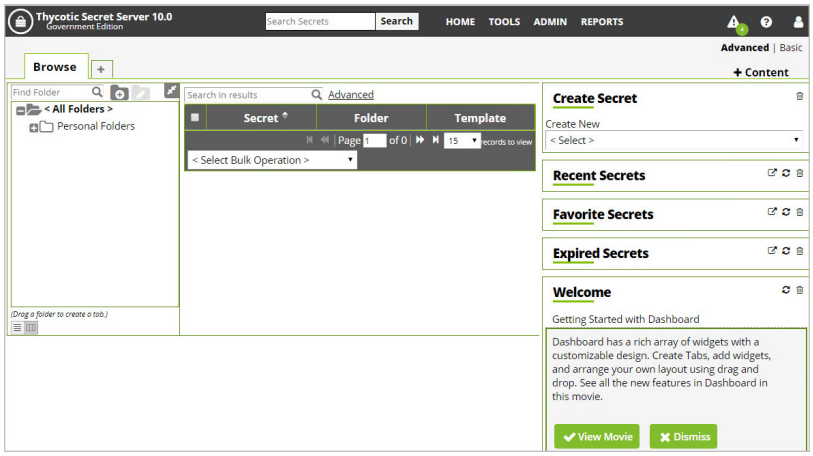
*For information on setting up your Secret Server Local Administrator account, see section 3.2.

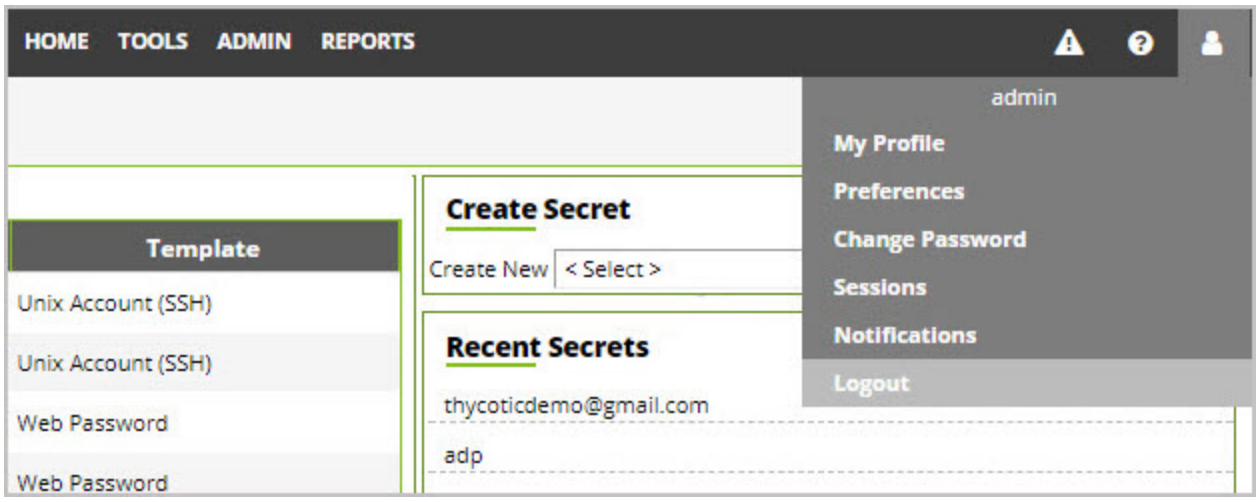
After enforcing the Login Policy Banner (for instructions, see section 5.4.1), the login screen will appear this way:



Once user accounts are configured, both Local accounts and Active Directory users will access Secret Server through this login screen. To configure Active Directory users, see section 7.1.

You will be directed to Secret Server's landing page for Administrator accounts.



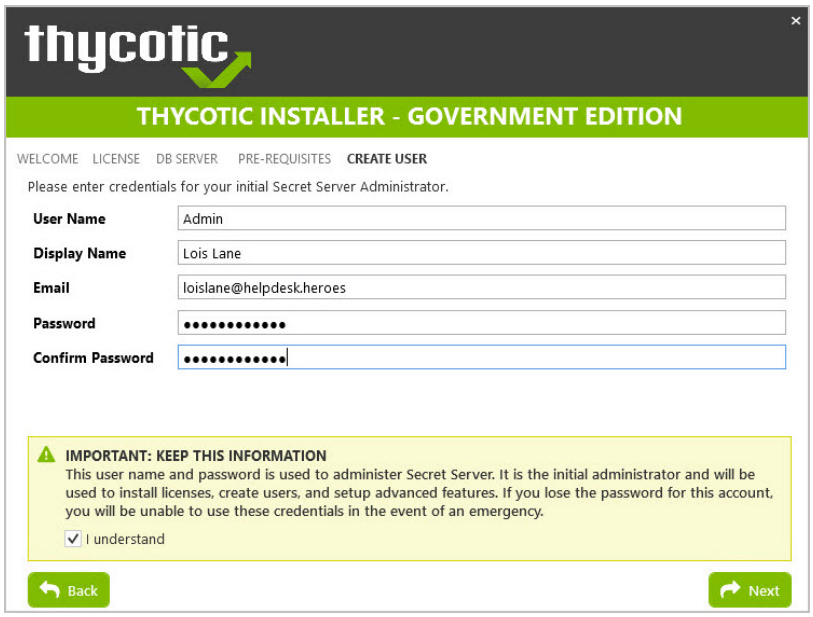


To logout, navigate to the **profile icon** in the upper right-hand corner and select **Logout** from the dropdown list. Once user accounts are created in Secret Server from the Admin profile, users can login using user-level credentials by navigating to your organization's login URL.

More detailed information for setting up Secret Server including customizing Secret Server your dashboard and creating users can be found in Secret Server's documentation.

Local Admin Account

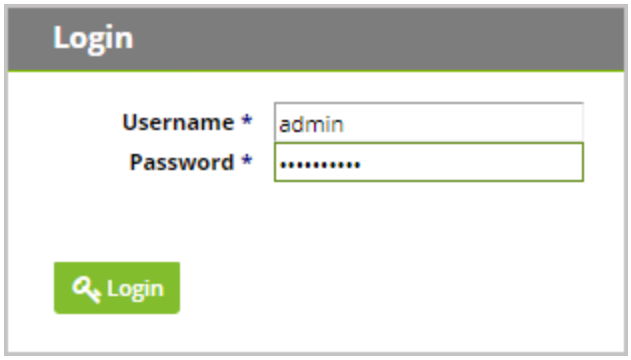
To create Secret Server's Local Administrator Account you will be prompted for information through the installer (step 7 in section 2.3.2 of this guide). Save the information you provide during this step so that you can login to the administrator account when the installer finishes setting up your instance.



Installing License Keys

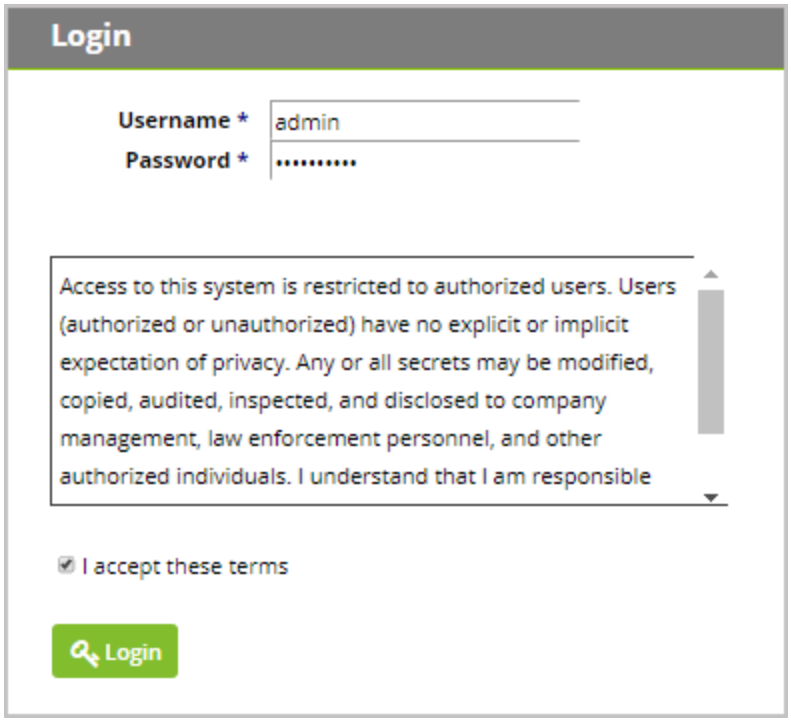
After logging into Secret Server with Secret Server's local administrator account, the first thing you will need to do is install your organization's license keys. You will need three different license keys, one for the Secret Server Edition being used, one for Support, and one for Users. Install your keys by navigating to **Admin | Licenses**, click **Install New License**. To install multiple licenses at once, click the **Bulk Entry Mode** button. Add your licenses according to the example format and click **Add Multiple Licenses**, enter the required informational fields included in your trial or purchase license email and then click **Activate**.

To activate newly installed licenses, click **License Activation** to log in.



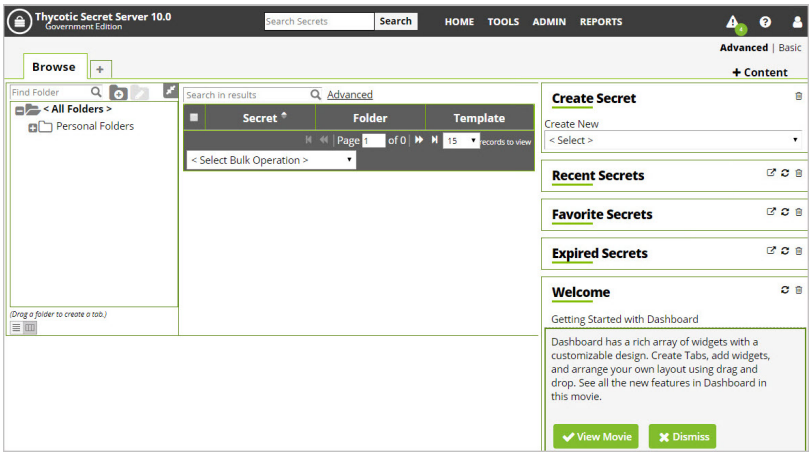
*For information on setting up your Secret Server Local Administrator account, see section 3.2.

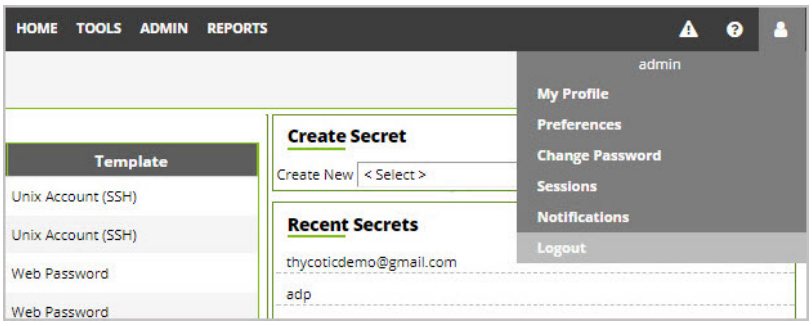
After enforcing the Login Policy Banner (for instructions, see section 5.4.1), the login screen will appear this way:



Once user accounts are configured, both Local accounts and Active Directory users will access Secret Server through this login screen. To configure Active Directory users, see section 7.1.

You will be directed to Secret Server's landing page for Administrator accounts.



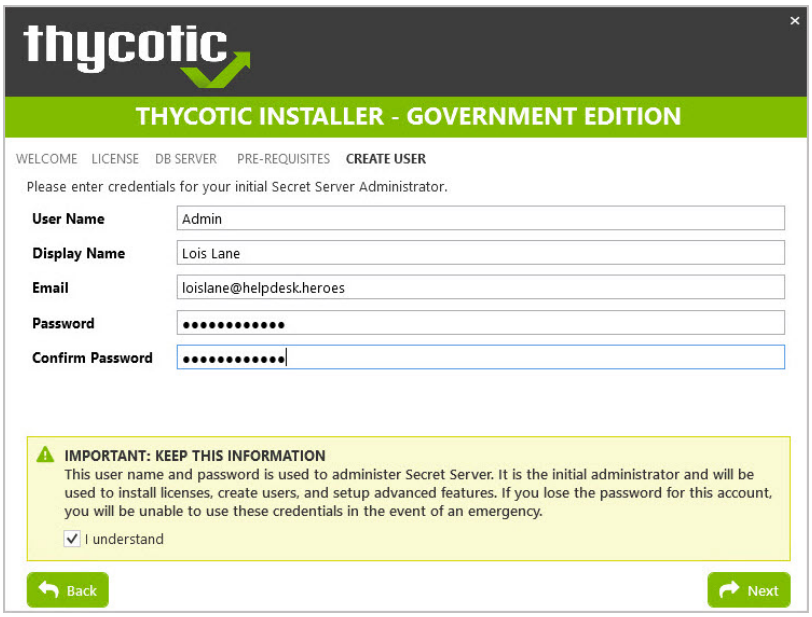


To logout, navigate to the **profile icon** in the upper right-hand corner and select **Logout** from the dropdown list. Once user accounts are created in Secret Server from the Admin profile, users can login using user-level credentials by navigating to your organization's login URL.

More detailed information for setting up Secret Server including customizing Secret Server your dashboard and creating users can be found in Secret Server's documentation.

Local Admin Account

To create Secret Server's Local Administrator Account you will be prompted for information through the installer (step 7 in section 2.3.2 of this guide). Save the information you provide during this step so that you can login to the administrator account when the installer finishes setting up your instance.



Installing License Keys

After logging into Secret Server with Secret Server's local administrator account, the first thing you will need to do is install your organization's license keys. You will need three different license keys, one for the Secret Server Edition being used, one for Support, and one for Users. Install your keys by navigating to **Admin | Licenses**, click **Install New License**. To install multiple licenses at once, click the **Bulk Entry Mode** button. Add your licenses according to the example format and click **Add Multiple Licenses**. button, enter the required informational fields included in your trial or purchase license email and then click **Activate**.

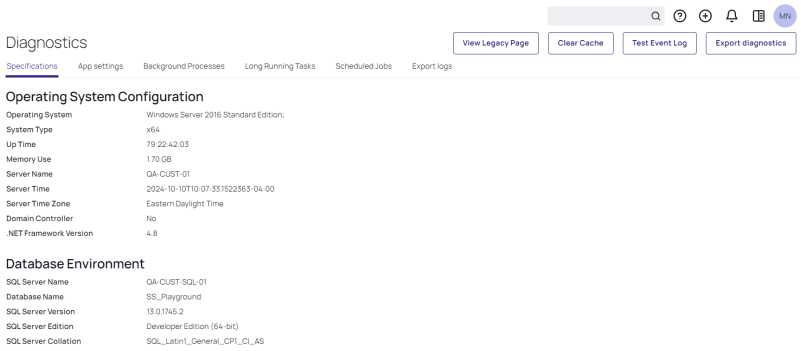
To Activate newly installed licenses, click the **License Activation**

Updating Secret Server

Checking Software Version

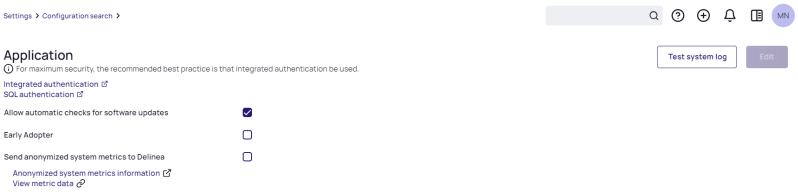
Verify Software Version and Components

A shorthand representation of the software version appears in the upper left-hand corner of any user screen, along with the product edition. The full software version number is listed in the **Secret Server Environment** section by navigating to **Admin > Diagnostics**:



Downloading Software Updates

By default, Secret Server is configured to notify you immediately when software updates are available. To customize this behavior, navigate to **Admin > Configuration**, select **Application** from the General section, and click **Edit**, then check the box next to **Allow Automatic Checks for Updates**. This setting will create a banner at the top of your Administrator account's user screen when a new update is available for download. Disabling this feature will prevent automatic update checks and banner displays.



If your Secret Server web server has no outbound access, you can elect to receive updates through Delinea's Support Portal by logging into your Portal account at support.delinea.com, navigating to your Account Settings, and signing up for the Delinea Mailing List.

After receiving a notification for the latest software release, navigate to https://support.delinea.com/s/download-onprem to download the latest .zip file.

To perform an upgrade without outbound access, follow the steps in "Upgrading Secret Server Without Outbound Access" on page 149.

Configuring Authentication and Login

The way that Secret Server authenticates interactive users depends on the type Secret Server of authentication that your organization has configured. Secret Server can use both **Local Accounts** and/or Active Directory **Domain Accounts** for authentication into Secret Server.

Local Authentication

A local user account is stored and managed by Secret Server. To successfully authenticate a local user must login with a matching username and password. When using local login, user credentials are checked against the internal authorized users' database.

To **create**, **edit**, or **remove** a local user account you must navigate to **Admin | Users** and locate any users whose Domain is listed as "Local" or **Create New** users.

Domain Authentication

A domain user account is stored and managed by Secret Server, but subject to changes made in Active Directory. To successfully authenticate a user must login with an Active Directory account that exists in Secret Server with matching Secret Server credentials. When using domain login, the TOE (Secret Server) initiates an authentication request to the external domain controller (Active Directory) using LDAP over TLS, and only allows access after receiving a successful result message.

For more information on syncing Secret Server with Active Directory credentials see section **7.1 Configuring Active Directory Sync**

Account Lockout Configuration

To access any information or functionality within Secret Server, users must login with correct local or domain credentials. To comply with Common Criteria regulations, Secret Server must be configured to prevent repeated unsuccessful attempts at logging in. **Account Lockouts** are used for this purpose. Configurable by the Secret Server Administrator, an account becomes inaccessible after a limited number of unsuccessful authentication attempts until an Administrator unlocks the user's account.

To configure settings for Account Lockouts, navigate to **Admin > Configuration > Login** tab, then click **Edit** and adjust the number for **Maximum Login Failures**. Default for this setting is five attempts.

To Unlock a user's account, navigate to **Admin > Users > Select the User** and click **Edit**. Change the value for "**Locked Out**" from Yes to **No**, then click **Save**.

Lockout Window

Secret Server is programmed with a default Lockout Window of 60 minutes. This means that once a user has locked out their account, they will be able to login with the correct credentials after a period of one hour has passed. Account lockouts are designed to prevent brute force attacks.

How to Disable "Allow Remember Me" during Logins

By default, the Secret Server installer will disable the "Allow Remember Me" caching feature during logins, however, to ensure this is feature is disabled, navigate to **Admin > Configuration > Login** tab, and verify that the first setting "**Allow Remember Me**" is set to **No**. If this is set to Yes, **Edit** the page and uncheck the toggle, then **Save**. Audits for this setting are logged under **Admin > Configuration > General** tab, by clicking the **View Audit** button. To filter log results, search for "**AllowRememberMe**" in the search bar.

Configuring the Login Banner

To configure the Login Banner, navigate to **Admin > Configuration > Login** tab, scroll to the bottom of the page and click the **Login Policy Agreement** button, then **Edit** and check the **Enable Login Policy** and **Force Login Policy** boxes.

Enabling these boxes will 1) Reveal the **User's Policy Agreement** on the Login page, and 2) Force users to **Agree** to the policy when logging into Secret Server.

How to Modify the Login Banner Messaging

To modify the display text from this **Login Policy Agreement**, go to Secret Server's Web Server and open **File Explorer**. Navigate to a text file called "**policy.txt**" (default file path location is

C:\inetpub\wwwroot\SecretServer\policy.txt)

Open this file in Notepad and adjust the text according to your organization's policy requirements, then **Save**. The default text reads as follows:

Access to this system is restricted to authorized users. Users (authorized or unauthorized) have no explicit or implicit expectation of privacy. Any or all secrets may be modified, copied, audited, inspected, and disclosed to company management, law enforcement personnel, and other authorized individuals.

I understand that I am responsible for protecting the confidentiality of company secrets and will comply with the company Information Security Policy. Unauthorized or improper use of this system may result in administrative disciplinary action, civil and/or criminal penalties. By continuing to Login, I am indicating my awareness of and consent to these terms and conditions of use.

** *CLOSE THIS SITE IMMEDIATELY if you do not agree to these conditions* **

Configuring Session Timeouts

To configure Session Timeouts, navigate to **Admin > Configuration > General** tab, click **Edit**. Under the User Experience section, check the **Force Inactivity Timeout** check box, then adjust the number of minutes of inactivity before an active session in Secret Server will timeout and force users to login again. Click on the **Save** button to save your changes.

Configuring IP Address Restrictions

To configure IP address restrictions, navigate to **Admin > More… > IP Addresses** and click **Create New**. Then provide the **IP Address User/Network Name** and the **IP Address Range**, click **Save**.

IP address restrictions can then be set at a user-level by navigating **to Admin | Users** and clicking on a user name, then click **Change IP Restrictions** and click on the Restriction checkbox for the restriction(s) you would like to enable for that user. **Save** changes.

To restrict group access into Secret Server by a specific IP address or IP address range, simply configure your IP address or range as listed above, then navigate to **Admin > Groups > Create New**. Select all employees or groups of employees to impose this restriction on and move left into the **Members** box. Click **Save**. Next click the button **Change IP Restrictions** and check the box for your desired IP Address/Range. **Save** changes to apply. Now users will now be restricted from accessing Secret Server outside of the designated IP Address Range.

Managing Local Users

Create New Local Users

To create a new local user in Secret Server, navigate to **Admin | Users** and select the **Create New** button. Provide the requested information on the **Edit User** page.

Manage Local Users

Click on the **Username** for details about a specific local user. Secret Server will provide information including **User** and **Display Names**, **email address**, **Domain** affiliated, whether the user has **Two Factor** verification set up on their account, whether the user is **Enabled**, **Locked Out** from accessing their account, or whether the account is not a User account but an **Application Account**. The **View User** page also provides information about **Groups** and **Roles** for this user.

To edit these settings, click the **Edit** button. By editing a local user you may change their **display name**, **email address**, allow **Two Factor**, **disable/enable** the account, or **lock/unlock** the account.

You also can **create a new, one-time password** for the user's account. If you change a user's password they will be prompted for a new password after logging in with the Administrator-created password.

Managing Domain Users

Configuring Active Directory Sync

Secret Server can integrate with Active Directory by allowing users to use their Active Directory credentials to login to Secret Server. According to Common Criteria compliance, Active Directory relies **on LDAPv3 (RFC 2251) protocol**, which is not configurable by users.

In order to setup Active Directory in Secret Server, you will need to:

1. **Create a Sync Secret**
2. **Specify the domain to authenticate against**
3. **Configure TLS with Active Directory**
4. **Set Synchronization Groups**
5. **Turn on Active Directory Sync**

Secret Server relies on a primary "Sync" secret to connect to the LDAPv3 server in Active Directory. Once connection has been made to Active Directory through this secret, Secret Server needs to know which domain within Active Directory to authenticate against, and within that domain which specific Active Directory Group(s) to synchronize with. The Active Directory Sync in Secret Server targets Secret Server user account credentials from Active Directory.

> Secret Server will categorize users according to group information from Active Directory, but Secret Server does not create, delete, or alter Active Directory Group Policies.

Create a Sync Secret

Before synchronizing users, you must first create a secret to be used as the Sync Secret. This secret should contain **Domain Admin credentials** (or an account with appropriate permissions for **Read Access** to all your organization's AD objects).

From Secret Server's dashboard you can create this secret through the Create Secret Widget.

1. Next to "Create New," select **Active Directory Account** from the dropdown list.
2. Add a **Secret Name** and provide the **Domain Name**, **Username**, and **Password** for the Sync Secret that will be able to access Active Directory with Admin credentials. **Save**.

Specify the Domain & Enable Active Directory Integration

Specify which domains Secret Server will be able to authenticate against. Secret Server can synchronize with any number of domains.

1. Once logged into Secret Server – Click on **Admin** | **Active Directory**
2. From the AD Configuration page, click **Edit.**
3. Check **Enable Active Directory Integration** in the Active Directory Integration section
4. Select **Enable Synchronization of Active Directory** in the Active Directory User Synchronization section
5. Wait a few seconds for the screen to update
6. Next to **User Account Options** select **User status mirrors Active Directory (Automatic)** from the drop-down. This allows Secret Server to mirror any changes made to Active Directory automatically.
7. Click **Save**
8. Click **Edit Domains**
9. Click **Create New**
10. Provide the **Fully Qualified Domain Name**
11. Provide a **Friendly Name**
12. Ensure that the box next to **Active** is checked.
13. Check the box next to **Allow Logins From Domain**.
14. Select your Sync Secret by clicking **No Selected Secret**
15. Search for the secret created earlier.
16. Click **Save and Validate**.

The Active Directory Sync Secret will be used to synchronize users and groups, **it will require permission to search and view the attributes of the users and groups. If you plan on using Discovery (NOTE: Discovery is not under Common Criteria's scope), the account will also need permissions to scan computers on the network for accounts.**

Configuring TLS with Active Directory

To ensure that TLS is configured with Active Directory:

- From the **Admin > Active Directory** > **Edit Domains** > **Create New**page (continuing from previous section), after entering the requested information, click **Advanced (not required)** and check the **Use LDAPS** box to enable. Click **Save And Validate** to save this domain.

**If the TLS connection to Active Directory fails, the user will be notified and the failure will be logged. Secret Server does not automatically retry to connect to TLS but will retry the next time a user attempts to connect to AD.*

More information for setting up Active Domain with LDAPS can be found at

https://blogs.msdn.microsoft.com/microsoftrservertigerteam/2017/04/10/step-by-step-guide-to-setup-ldaps-on-windows-server/**"We're no longer updating this content regularly."**

Setup Synchronization Groups

Once the domain has been added, click the "**Edit Synchronization**" button on the Active Directory Configuration page.

The Available Groups represent all accessible groups on the specified Active Directory domain. The user membership can be previewed with the Group Preview control.

- Select the desired group from the Available Groups that contains the Active Directory accounts for users you would like to create in Secret Server and move it into the **Synchronized Groups** area, click **Save**. This allows you to tailor specific Active Directory users to have domain accounts in Secret Server.

Managing Domain Credentials

Updating Domain Credentials

Updating Active Directory (AD) Credentials and passwords happens directly through Active Directory and syncs with Secret Server according to a schedule. You can synchronize your Active Directory accounts at any time by navigating to **Admin | Active Directory** and clicking the **Synchronize Now** button. *You cannot add or edit Active Directory credentials through Secret Server directly*.

When a user logs into Secret Server using Active Directory credentials or uses an AD account to launch a session, the credentials are sent to the Domain Controller in real time for authentication verification. Therefore, if an AD user account has been updated or removed, changes will be reflected immediately in Secret Server.

Failed Domain Authentication

If the connection between Secret Server and the AD domain breaks, domain users Secret Server will fail to authenticate into Secret Server until the connection is re-established. Secret Server will log all failed authentication attempts by users.

To find these audit logs, navigate to **Admin > Users >** select a user **> View Audit** or refer to section **12.0 Local Auditing**.

Common Criteria Roles and Permissions

User Roles can be edited by navigating to **Admin | Roles** and choosing **Assign Roles**, then **Edit**. On the Role Assignment page you will be able to select which users should be assigned to which role.

The **User Roles** that comply with Common Criteria standards are:

- **Administrator**
- **User**
- **Read Only User**

By default, Administrators have all possible Role Permissions. To view the list of permissions and security attributes that associate to each User Role, navigate to **Admin | Roles** and select each role to view the list of permissions. Each user session will be limited by the Role Permissions assigned to that user.

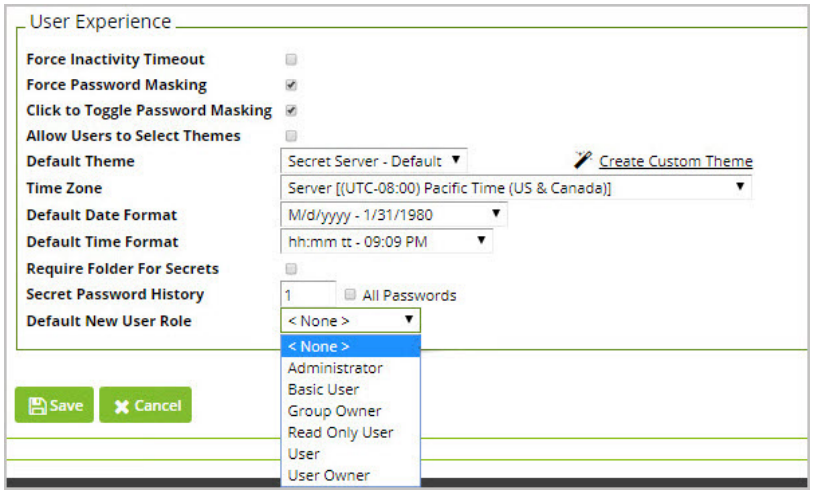
Each can be associated with individual user identities, or can be associated with either a local Group or an Active Directory Group. To assign a Role to a User or Group, navigate to **Admin > Roles > Assign Roles**.

To administer configurations required for Common Criteria standards, an Administrator requires the default permissions that are included for the Administrator Role.

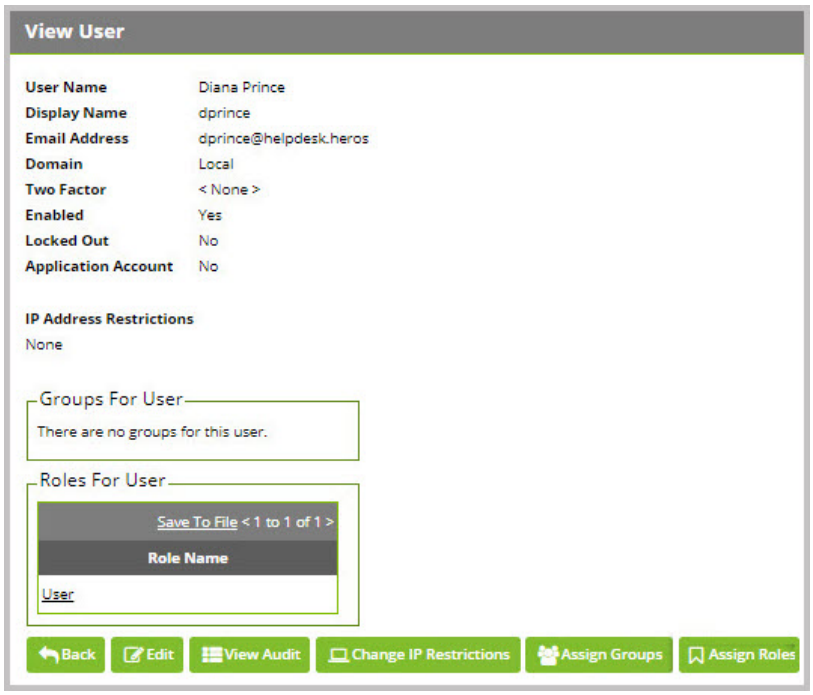
Assigning Roles to Users

The default role assignment for new users is set by navigating to **Admin | General** tab **| User Experience** section (scroll to bottom). To ensure that no new users are created or imported with any extra privileges, make sure that this setting is set to “None.”

To edit this default setting, click **Edit** and select “**None**” from the dropdown list as shown by the screenshot below. Then click **Save**.



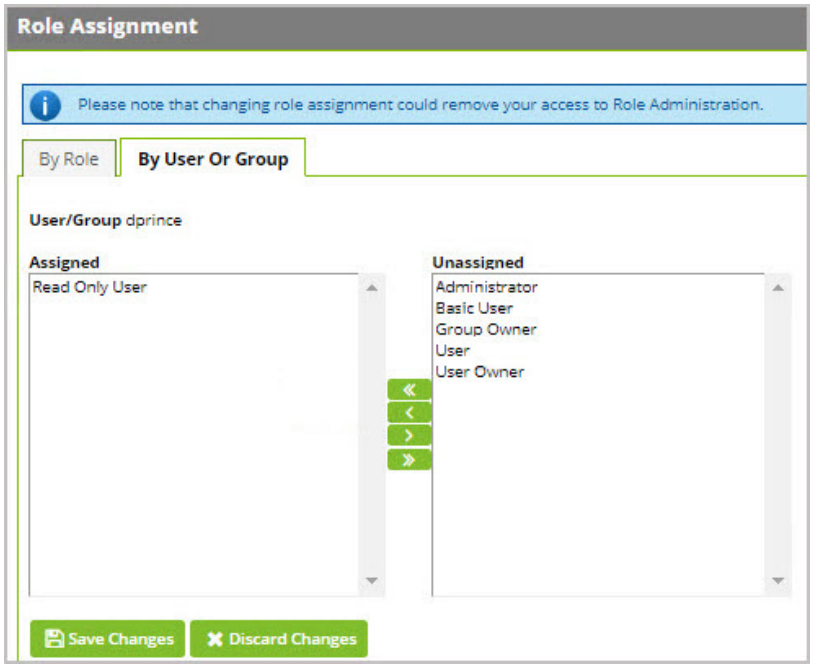
To assign users to Roles that are compliant with Common Criteria standards to specific users or groups, navigate to **Admin | Users** and select the user, OR navigate to **Admin | Roles;** then click **Assign Roles.**



Under the **By User Or Group** tab, select a user or group from the dropdown, then click **Edit**.

The Roles supported in Common Criteria include **Administrator, User,** and **Read Only User**.

To apply one of these roles, select it from the right-hand list and move it to the left side under the Assigned box. Ensure that every user only is assigned only one of these Roles. If both “Administrator” and “Read Only User” are assigned to the same user, the user will maintain full Administrator access to Secret Server. Click **Save Changes**:



Management Functions Based On Role

This section describes management activities and corresponding roles of the evaluated security functionality.

Role	Management Functions
Read-only User	Search and list Secrets
User	Use Secret/Launch session
User	Request access to Secret
Administrator	Create, view, expire, edit, and assign Secrets
Administrator	Perform bulk operations on Secrets
Administrator	Create and manage groups
Administrator	Create and manage roles, assign roles to users
Administrator	Create and manage containers (folders)
Administrator	Create and manage Secret policy
Administrator	Configure TOE SF (see Table 5)
Administrator	Create, manage, and unlock local accounts
Administrator	Configure IIS, SQL, syslog
Administrator	Update TOE

Table 5: Management Functions and Roles

By default each User Role is attached to Permission Sets. To view the specific permissions that each role is attached to, navigate to **Admin | Roles** and click into the user roles listed to see the list of permissions.

Organizations can tailor these user roles to maintain whatever permissions settings are required for your specific user environment.

8.3 Common Criteria Management Activities Based On Role

The following table specifies the user role required to allow each management activity listed according to Common Criteria Standards.

Requirement	Management Activities	Role
ESM_EAU.2	Management of authentication data for both interactive users and authorized IT entities (if managed by the TSF)	Administrator

Requirement	Management Activities	Role
ESM_EID.2	Management of authentication data for both interactive users and authorized IT entities (if managed by the TSF)	Administrator
ESM_ICD.1	Definition of identity and credential data that can be associated with users (activate, suspend, revoke credential, etc.)	Administrator
	Management of credential status	Administrator
	Enrollment of users into repository	Administrator
ESM_ICT.1	Configuration of circumstances in which transmission of identity and credential data (and object attributes, if applicable) is performed	Administrator
FAU_STG_ EXT.1	Configuration of external audit storage location	Administrator
FIA_AFL.1	Management of the threshold for unsuccessful authentication attempts	Administrator
	Management of actions to be taken in the event of an authentication failure	Administrator
FIA_USB.1	Definition of default subject security attributes, modification of subject security attributes	Administrator
FMT_MOF.1	Management of sets of users that can interact with security functions	Administrator
FMT_SMR.1	Management of the users that belong to a particular role	Administrator
FTA_SSL.3	Configuration of the inactivity period for session termination	Administrator
FTA_TAB.1	Maintenance of the banner	Administrator
FTP_ITC.1	Configuration of actions that require trusted channel (if applicable)	Administrator
FTP_TRP.1	Configuration of actions that require trusted path (if applicable)	Administrator

Table 6: Management Functions and Roles by Common Criteria Requirement

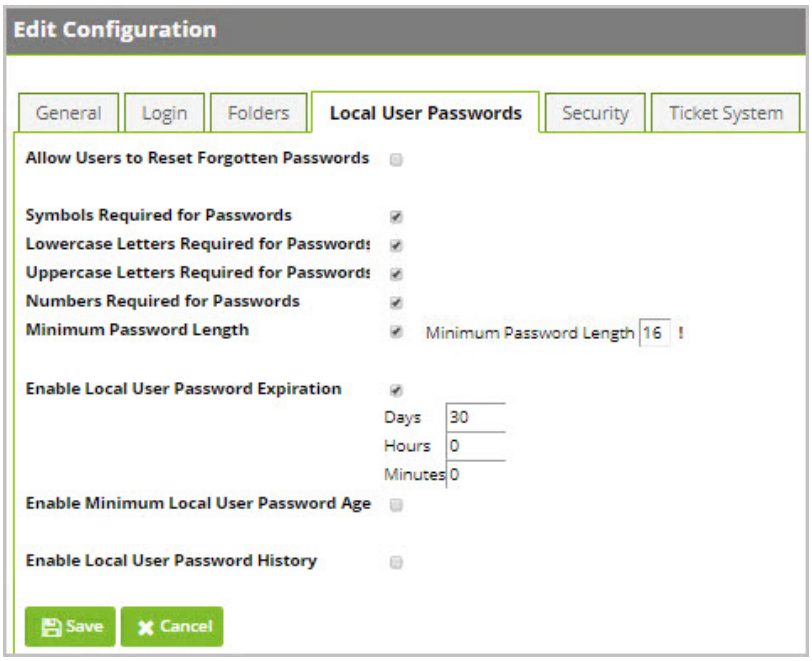
Managing User Passwords

Password Requirements for User Authentication

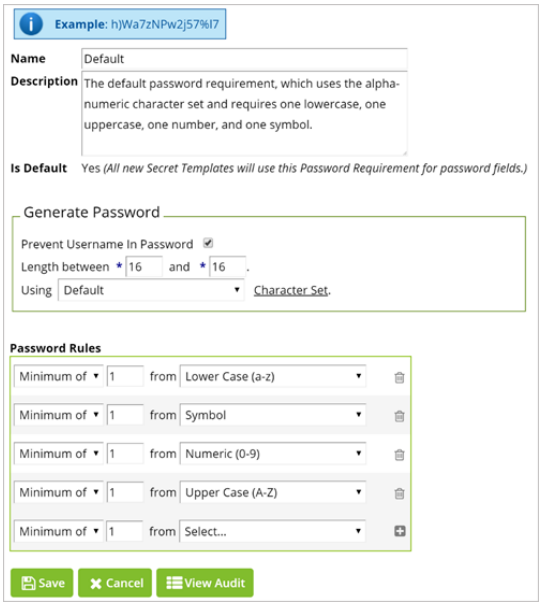
The Secret Server password policy for your user accounts is determined either through Active Directory or through Local User Passwords settings. To configure settings for Local User Passwords, navigate to **Admin > Configuration > Local User Password** tab and click **Edit**.

To meet Common Criteria Compliance regulations, passwords must be a **minimum of 16 characters** and include any subset of the following requirements (as demonstrated below):

- Upper case letters
- Lower case letters
- Numbers
- These Special Characters: ! @ # $ % ^ & * ( )



Passwords in Secret Server are randomly generated according to admin-defined password complexity policies. By default the generator requires at least one upper case letter, one lower case letter, one number, and one symbol. The number of each type of character can be modified, and custom character sets can be created and used in password policies.



Resetting User Authentication

To reset an Active Directory user account password, you must go *through Active Directory*.

To reset a Local User Password, follow these steps:

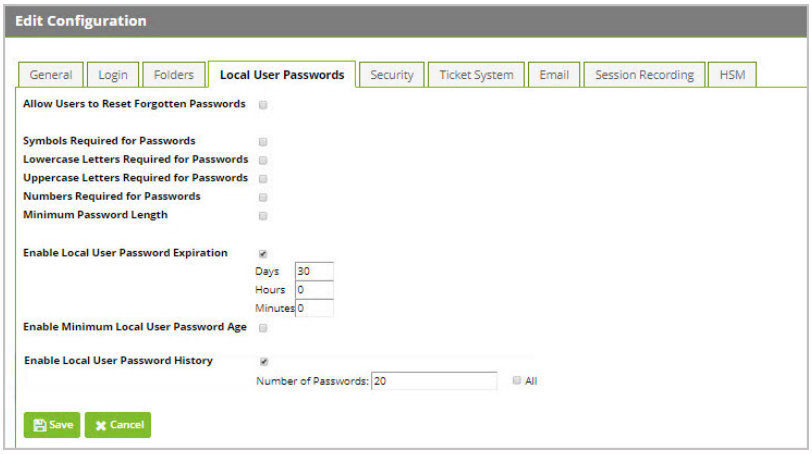
1. Navigate to **Admin | Users**.
2. Select the user who needs a reset.
3. Select **Edit**.
4. Type a new password twice. This is a *temporary password* that you must provide to the user. Immediately after the user logs into Secret Server they will be prompted to change their password.

If you are locked out of the Secret Server Local administrator account and you cannot request a reset through a linked administrator email account, contact the Delinea Support Team to request a password reset, and have your organization's **security pin code** at hand.

Setting Local User Password History Requirements

Ensure that local users cannot re-use old passwords to an administrator-settable number of past passwords used by that user.

1. Navigate to **Admin > Configuration > Local User Passwords**
2. Click **Edit**.
3. Check the toggle box next to **Enable Local User Password History**.
4. Set the number of historic passwords you would like to block for Secret Server Users.
5. Click the toggle next to **All** to block users from ever re-using any previous password.
6. Click **Save** to save changes.

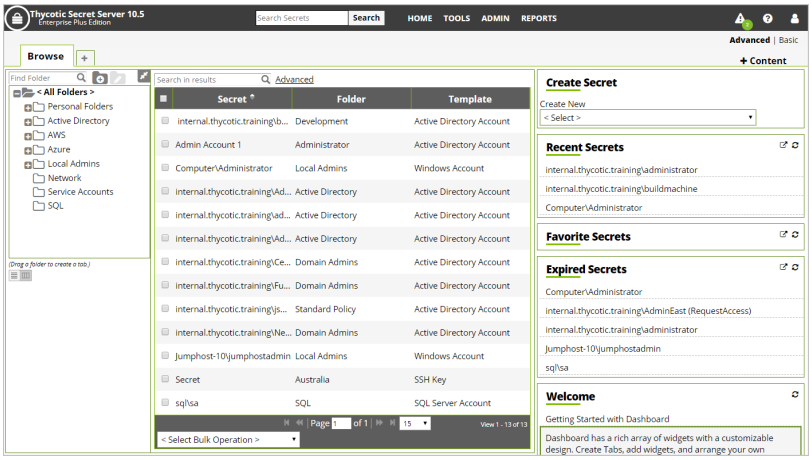


Managing Secrets

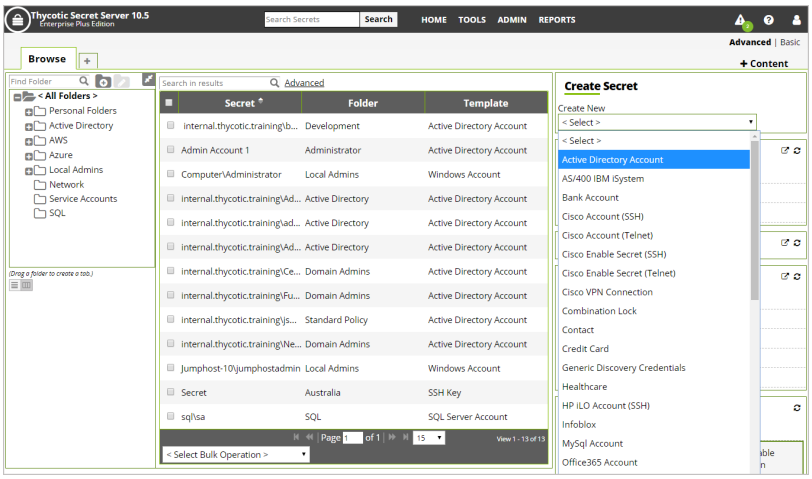
Creating Secrets

Each object that is stored within Secret Server is referred to as a **Secret**. Usually, a Secret will be a username and password combination. Other examples of Secrets can include SSH keys, contact information, or safe combinations.

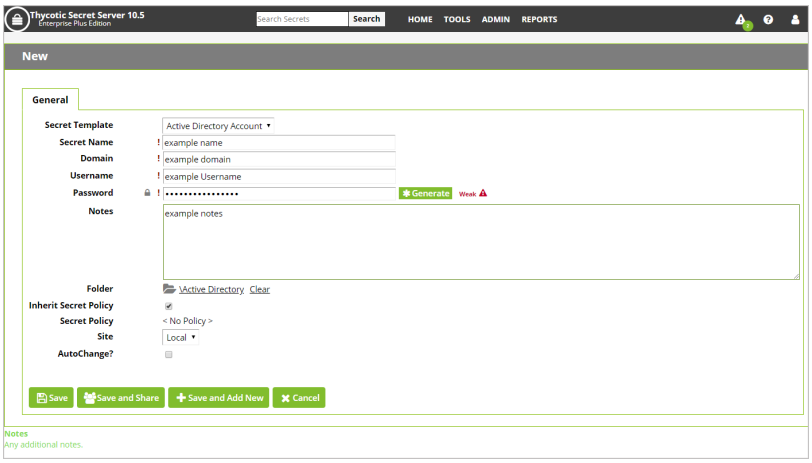
To create a Secret, use the “Create Secret” widget.



Select the type of Secret you want to create.



Enter the data for the Secret you want to save, and make sure to choose the Folder that the Secret will be saved into. Click **Save**, and your Secret has been created.



Configuring Secret Templates

Secret Server manages privileged account credentials through a highly configurable system of secrets. **Secret Templates** are used to create secrets and define object attributes for secrets. Templates can be configured according to account requirements.

The following is a list of Secret Templates that are compliant with Common Criteria standards available in the Government edition of Secret Server:

- **Active Directory Account**
- **Bank Account**
- **Combination Lock**
- **Contact**
- **Credit Card**
- **Password**
- **Pin**
- **Product License Key**
- **Security Alarm Code**
- **Social Security Number**
- **Unix Account (SSH Key Rotation)**

To follow Common Criteria standards, navigate to **Admin | Secret Templates** and click the **Active Templates** button. Ensure that only the templates listed above are selected in the Active column. Then **Save**. Users only can create Secrets using templates marked as Active.

To view object attribute data in a template from the dropdown list of Active Templates, **select a template** from the dropdown list and then click **Edit**.

Each of the listed templates below only hold object attribute data that are defined locally by Secret Server. The objects, secrets, and templates are as follows:

Object

Attribute

Objects in Secret Server are defined by attribute data. All object attribute data is locally defined by Secret Server.

Secret

Attributes Inherited from Template

Secret templates dictate the fields each secret contains, the launchers for each secret, and the remote password changer used. They also provide a default expiration, which can be changed on a per-secret basis.

Command Restrictions

Command restrictions are limited to SSH sessions and allow administrators to create multiple-choice command menus that users can follow. If SSH command menus are enabled, users cannot issue commands directly to the target system.

Field Data

Each field in a secret template is either “Text,” “Password,” or “Notes.” Password fields can be hidden from users and are updated when a password change has occurred on that secret’s account. Text fields are the standard field type and may include information such as “Domain” or “Username.”

Folder

Folders in Secret Server store individual secrets. Permissions applied to folders dictate which users can view the secrets inside that folder and which users can see the folders.

Password Requirements Rule Override

Password requirements can optionally be enforced on a per-template basis. If the password requirement is not enforced, users can manually type any password they want and will be notified if they enter a weak password that does not meet the password requirements. If the requirements are enforced, users cannot save the new password until it meets the requirements.

Policy Identifier

Secret policies are a collection of security and password settings that are applied to individual secrets or folders. Each setting of a secret policy can be configured as either default or enforced. Default allows users to later change the setting. Enforced locks the settings and cannot be modified on a per-secret basis unless the secret is moved out of the folder that has the secret policy attached. Secret policy settings include items such as “Remote Password Changing Auto Change,” and “Requires Approval for Access.”

Subject Identifier

The secret template of each secret identifies the type of store password or other data. This allows users to see the intended usage of each template.

Secret Name

The secret name is the label or title that describes the content of each secret. When Secret Server creates secrets automatically, the default naming convention is “host/account,” or “domain/account.”

Template

Field Parameters

Field parameters include username, password, and type. Secrets include a combination of field parameters that vary with user input. Secret type is defined by the template for each secret, such as “Bank Account” or “Active Directory Account.” The number of field parameters also are defined at the secret template level.

Password Change Policy

Password rotation is enabled or disabled on a per-secret or per-template basis.Password rotation frequency depends entirely on the expiration period for each secret. The default expiration time is 30 days for all secret templates.

Password Strength Policy

A password policy is a set of instructions on how each new password is created.Administrators must choose a minimum and maximum password length, which character sets are used, and how many characters from each set are required. See section 9.0 for details on setting password requirements.

Secret Expiration Policy

When a secret reaches its expiration date, it is flagged as “Expired.” If automatic password rotation is enabled for that secret, expiration triggers a remote password change. Expiration can be changed on each secret, but the default expiration period is set at the secret template level.

Secret Name Pattern

When Secret Server names a secret via the discovery import process, it uses the naming convention “hostname/username,” or “domain/username.” This is not enforced, so users can name secrets whatever they want.

Template Description

Secret template descriptions allow administrators to describe the purpose of a template when they create a new template, which is a best practice to avoid confusion.

Template Name

The secret template name is what users see in the drop-down menu when they create a new secret. It is also on the Secret Browse page so that users can see which secret template is associated with each secret.

Template Status

Like secrets, users can disable secret templates to make them invisible to users, unless they chose to view inactive templates. The status is either “enabled” or “disabled.”

Secret Access Policy

If users have view (or greater) rights, they can see whether a secret exists and can open the secret to view its data. If a user does not have view rights, the secret is invisible.

Secret Modification Policy

Each secret carries individual access permissions that are typically inherited from the secret's folder. These permissions determine which users can view the secret and which users can edit the secret's data. A user can have view, edit, or owner permission. With the view permission, a user can view but not modify a secret. The edit permission allows the user to modify field data. The owner permission allows users to grant or revoke access to other users.

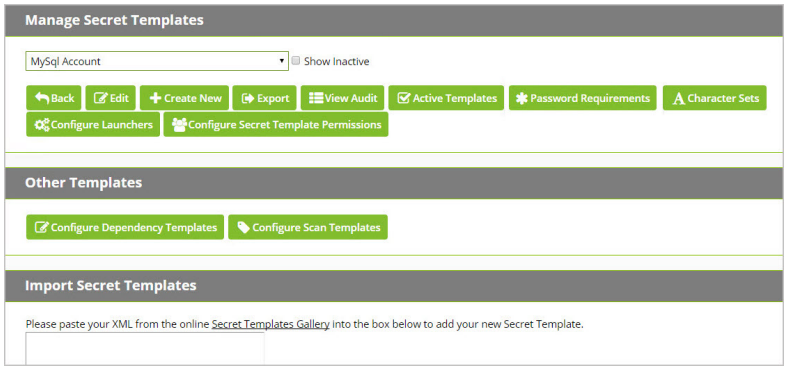
Configuring Password Policy for Secret Templates

When creating and rotating passwords for secrets inside of Secret Server, it is important to uphold strong requirements and to use Secret Server to manage changing requirements effectively.

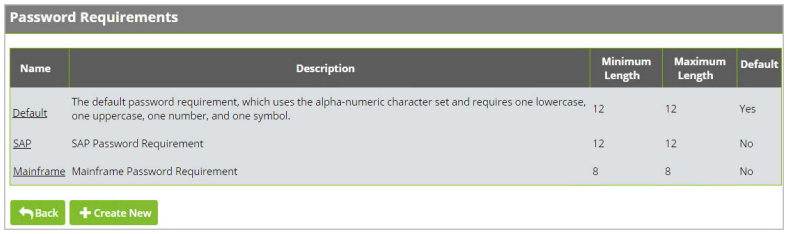
For example, Secret Server allows administrators to set the minimum password length to 6 characters, observe that a 7-character password and a 16-character password are both accepted, then change the minimum length to 8, observe that a 7-character password is then rejected but that a 16-character password is accepted.

In Secret Server password requirements can be set and applied *at the Secret Template level*. To adjust requirements:

1. Navigate to **Admin | Secret Templates**.

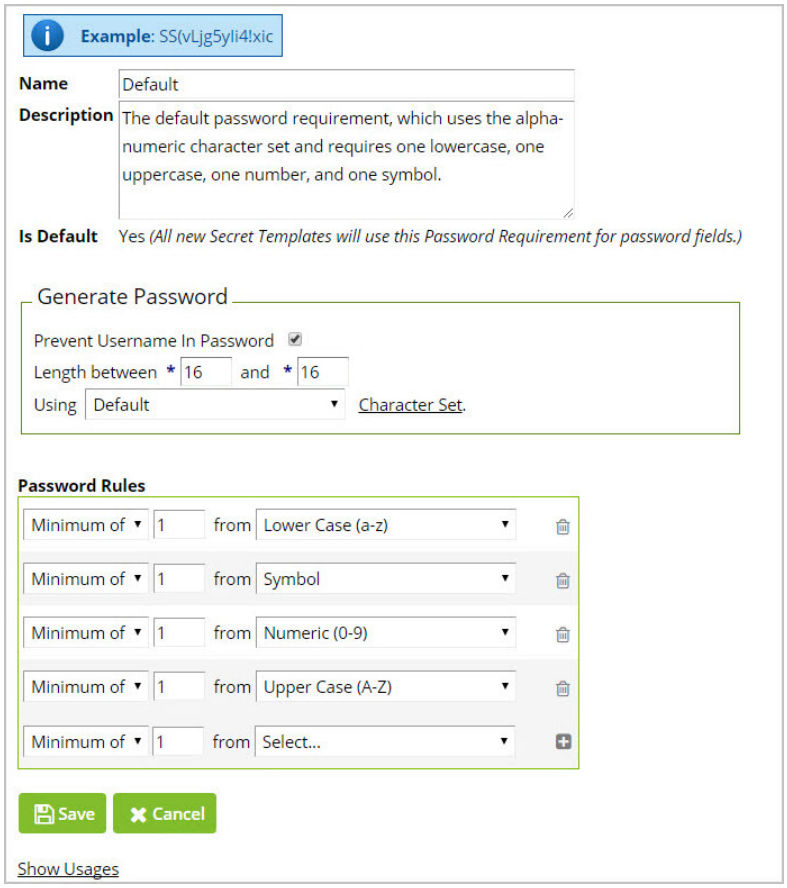


2. Select the **type of template** you want to adjust from the dropdown menu.
3. Click **Password Requirements** to open the Password Requirements" page.



- To edit *all* your organization's default password requirements on secret templates, select the **Default** password requirement from the list.
- To create a new password requirement specific to this template, select **Create New**.

4. Adjust the **Password Length** and **Character Set** requirements to the needs of your organization. You may assign your new requirement to any Secret Template or templates.



> Password Requirements set for these templates will be enforced for both human-generated and auto-generated secrets.

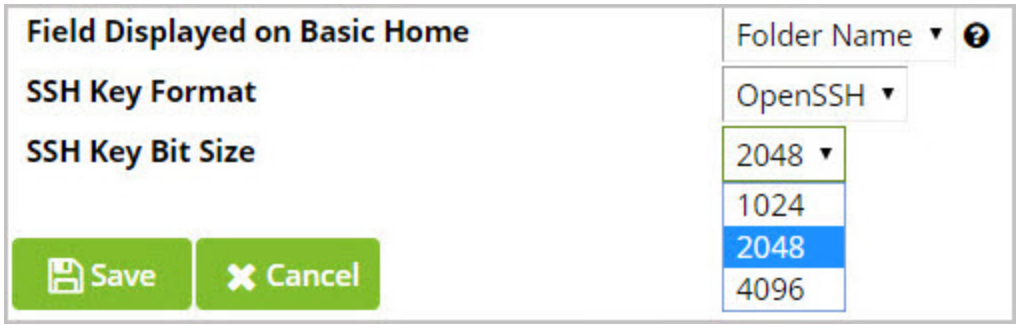
Authentication Strength for Non-Password Credentials

Secret Server uses RSA keys of 2048 bits or higher for secure authentication. These SSH keys are non-password credentials that can be managed by Secret Server. To ensure that these are maintained up to encryption standards, make sure that your **Unix Account (SSH Key Rotation)** Template is configured with an SSH Key Bit Size of 2048 or higher.

By default, the Government Edition of Secret Server will set this Template setting to 2048.

To adjust the bit size:

1. Navigate to **Admin | Secret Templates**.
2. Select **Unix Account (SSH Key Rotation)** from the dropdown list.
3. Click the **Edit** button.
4. From the Secret Template Designer page, click **Edit** to adjust settings. You can increase the bit size to 4096 from the default setting of 2048 if you choose, but do *not* lower this setting to 1024.



To create, store, and manage SSH keys in Secret Server, users must engage this Unix Account (SSH Key Rotation) Template. That means an SSH key will be created only when a standard encryption is enforced.

Configuring Remote Password Changing for SSH Key Rotation

Security Overview for SSH Key Rotation and PuTTY Launcher

SSH Key Rotation allows you to manage your Unix account private keys and passphrases as well as their passwords. The public/private key pair is regenerated and the private key is encrypted with a new passphrase any time a secret's password changes, either manually or automatically. The public key is then updated on the Unix machine referenced on the secret.

To use default SSH Key Rotation commands, the machine being managed must meet the following minimum requirements:

- SSH Key logins in OpenSSH format should be enabled on the target using keys. A secret can be created with keys in PuTTY format but they will be converted to OpenSSH when the key is rotated.
- Public keys should be stored in [~userhome]/.ssh/authorized_keys (not authorized_keys2).
- Grep and Sed should be installed on the target.
- If doing a privileged SSH Key Rotation, where a privileged user sets the key for another user, the privileged user must have sudo permissions that do not prompt for a password, as well as permissions to edit the user's authorized keys file with sudo.

If a system does not meet these requirements it may still be possible to do key rotation by modifying the key rotation command sets.

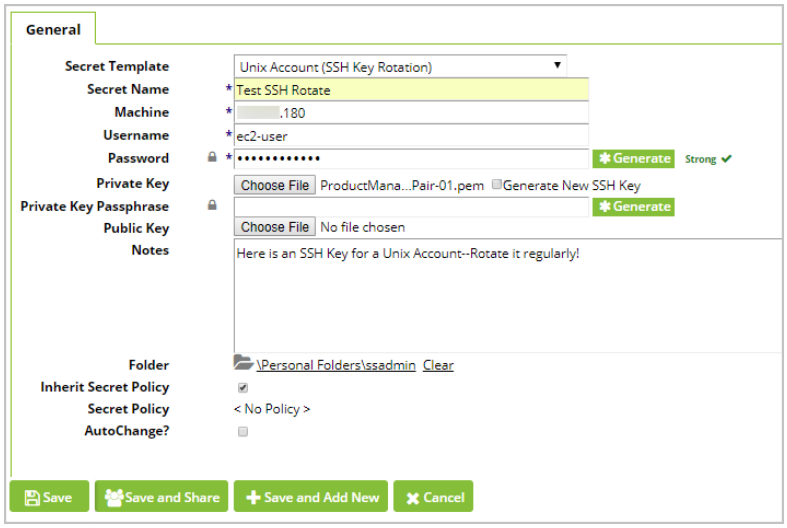
Creating a Unix Account (SSH Key Rotation) Secret

Under Secret Server's Common Criteria compliance standards, you can set up Secret Server to rotate SSH Keys for Unix Accounts.

To setup a Launcher, you will need a **Unix Account (SSH Key Rotation)** Secret that is connected to a remote machine

To create a Unix Account (SSH Key Rotation) Secret:

1. From the **Home** Dashboard select the **Unix Account (SSH Key Rotation)** Template from the **Create Secret** widget.
2. Enter a **Secret Name**, **Machine Name**, **Username**, and **Password** for the Unix/Linux Account.
3. Select the SSH **Private Key** by browsing to it on your machine.
4. Enter the **Private Key Passphrase** for the SSH Key. If there is a corresponding Public Key, upload that as well.
5. Specify which Secret Server folder you want to store the Key in if it differs from the default folder path. Likewise, determine what secret policies you want to assign to this SSH key, if any.



To automatically rotate the private and public SSH key pairing upon clicking **Save**, click **Generate New SSH Key**. This action is a security measure to ensure that no one can access your SSH key unless they are doing so through Secret Server's vault.

Enabling the Launcher

By default, the Launcher is enabled.

To verify this, click on **Admin | Configuration**. Check the **Enable Launcher** Setting to ensure that this is set to **Yes**.

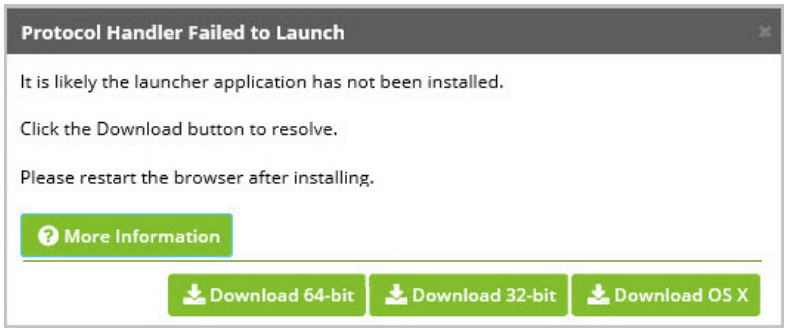
Using the Launcher

Note: SSL Certificates must be properly configured or the Launcher will fail to connect. So when you try to connect to Secret Server you receive a certificate error such as host name mismatch, you must resolve the cause of the error before the putty launcher will function. To directions on setting up certificates, see section **CREATE AN INTERNAL LINK** 11.2 Configuring X.509v3 Certificates.

1. From your Home screen, click the SSH secret
2. Click **View Secret**.

3. Click the **PuTTY Launcher** icon.

> The first time you perform this task you will receive a "**Protocol Handler Failed to Launch**" message. Select the type of launcher you need and **Run** the .msi file. Secret Server will download a very small process called a Protocol Handler that facilitates the connection between your machine and the endpoint. Once the Protocol Handler is downloaded, close out the "Failed to Launch" window and *refresh your browser page*.

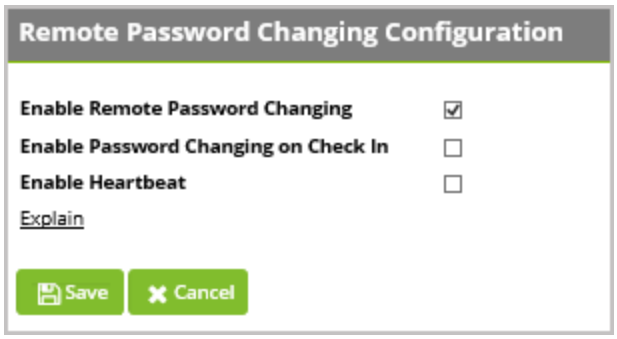


4. Click the launcher icon on the secret, and provide the machine name if prompted. The credentials will be passed along to the launcher and will open the appropriate launcher PuTTY session.

Enable Remote Password Changing

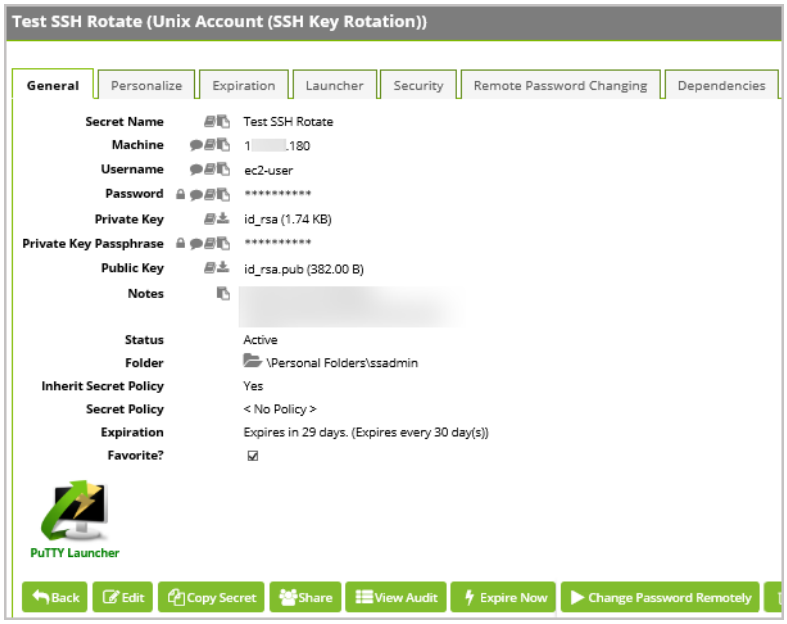
> To maintain compliance with Common Criteria regulations, Remote Password Changing is applicable for SSH Key Rotation ONLY.

1. Navigate to **Admin | Remote Password Changing**.
2. Click **Edit**
3. Check to select **Enable Remote Password Changing**.
4. Click **Save**.



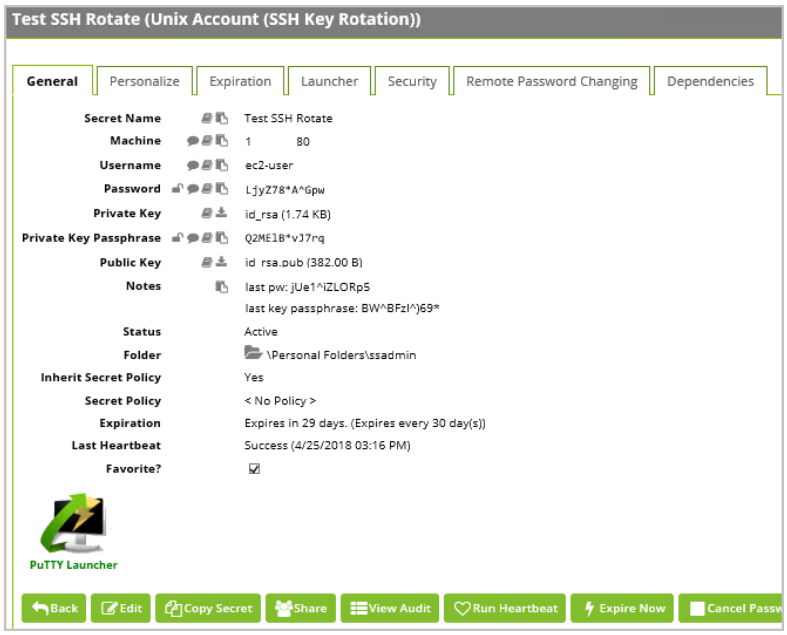
Rotate SSH Key Remotely

1. Navigate back to the SSH Key Rotation Secret's **View** screen.
2. At the bottom of the screen, click the **Change Password Remotely** button.



3. From the Change Password Remotely screen, **Generate** a new Password and Passphrase for your new SSH Key.
4. Next to Generate New SSH Key, leave the toggle checked.
5. Click **Change**. You will be directed to a **Password Scheduled for Change** screen.
6. Click **Back** to return to your secret's Remote Password Changing tab.

7. Navigate to the **General** tab.
8. Verify that a new password is listed. In the screenshot below you can see the previous passwords listed in the Notes section, confirming that the rotation was effective.



9. Click the **PuTTY Launcher** icon to confirm that Secret Server can still connect to the unix/linux machine using the newly rotated SSH Key.

Secret Expiration

A core feature of Secret Server is Secret expiration. Any template can be set to expire within a fixed time interval. For a Secret to expire, a field must be selected as the target of the expiration. For example, a Secret template for Active Directory accounts might require a change on the password field every 90 days. If the password remains unchanged past the length of time specified, that Secret is considered expired and will appear in the **Expired Secrets** panel on either the Dashboard's Expired Secrets widget or the Home page.

Secret expiration provides additional security by reminding users when sensitive data requires review. This can assist in meeting compliance requirements that mandate certain passwords be changed on a regular basis. When expiration is combined with Remote Password Changing, Secret Server can completely automate the process of regularly changing entire sets of passwords to meet security needs.

Setting up Secret Expiration for the Secret template

To set up expiration on a Secret, you must first enable expiration on the template from which the Secret is created.

To enable Secret expiration for a Secret template, navigate to **Administration | Secret Templates**. In the Manage Secret templates page, select the template from the dropdown list and click the **Edit** button. In the Secret template Designer page, click on the Change link. On this subsequent page, check the **Expiration Enabled?** box. You can now enter the expiration interval (every x number of days) as well as the field on the Secret you wish to expire and require to be changed. The interval setting can be overridden for each individual Secret.

Enabling expiration for a template will enable expiration for all the Secrets that were created using this template.

Setting up Secret Expiration for the Secret

Now that expiration has been enabled for the template, Secret expiration is enabled for the Secrets that were created using that template as well as Secrets created in the future. The Expiration tab will appear on the Secret View page and requires the user to have Owner permission on the Secret. If you would prefer to set a custom expiration at the Secret level, you can adjust the interval of expiration for the Secret by clicking the **Expiration** tab in the Secret View page. In the Expiration tab, you can set the Secret to expire using the template settings (default), a custom interval, or a specific date in the future.

Forcing Expiration

To force expiration, navigate to the **Secret View** page. From there, click **Expire Now**. This will force the Secret to expire immediately regardless of the interval setting. The expiration date will read "Expiration Forced".

Resetting an Expired Secret

To reset an expired Secret, you will need to change the field that has expired and is required to change. For example, if the field set to expire is the Password field and the current Password is "asdf", then a change to "jklh" will reset the expiration interval and thus remove the expiration text on the Secret View page.

If you do not know which field is set to expire, you will need to go to the Secret template that the Secret was created from. Navigate to **Administration | Secret Template** and select the template. Click the **Edit** button and then on the next page, click the "**Change**" link. In the "Change Required On" textbox you will see the field that is set to expire.

AutoChanging an Expired Secret

Remote Password Changing (RPC) is enabled under the Administration, Remote Password Changing page. Click **Edit** to enable Remote Password Changing, Secret Heartbeat, and Secret Checkout. Once enabled, all Secret templates with RPC configured will be available to use RPC.

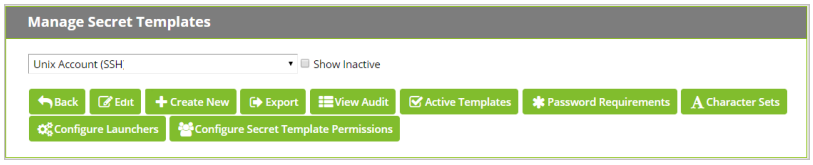
The Remote Password Changing tab contains the settings for configuring RPC on an individual Secret. Enabling AutoChange on a Secret will allow Secret Server to Remotely Change the Password when it expires. The user must have Owner permission on the Secret to enable AutoChange. When editing on the RPC tab, the Next Password field can be set. If left blank an auto-generated password will be used.

To auto-change passwords based on secret expiration leave the AutoChange schedule set to "**None**."

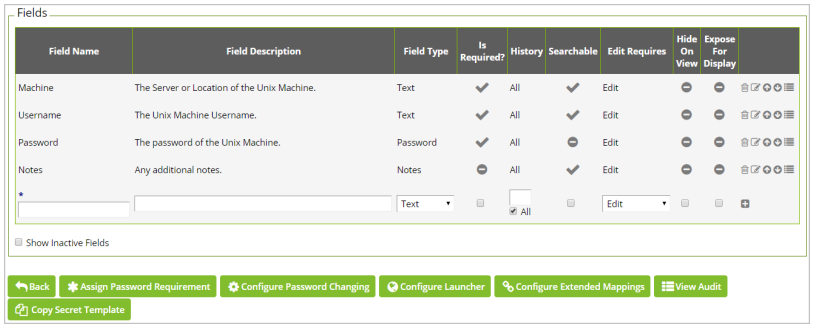
If the password change fails, Secret Server will flag the Secret as Out of Sync and continue to retry until it is successful. If the Secret cannot be corrected or brought In Sync, manually disabling AutoChange will stop the Secret from being retried.

Setting the Password Change Retry Interval

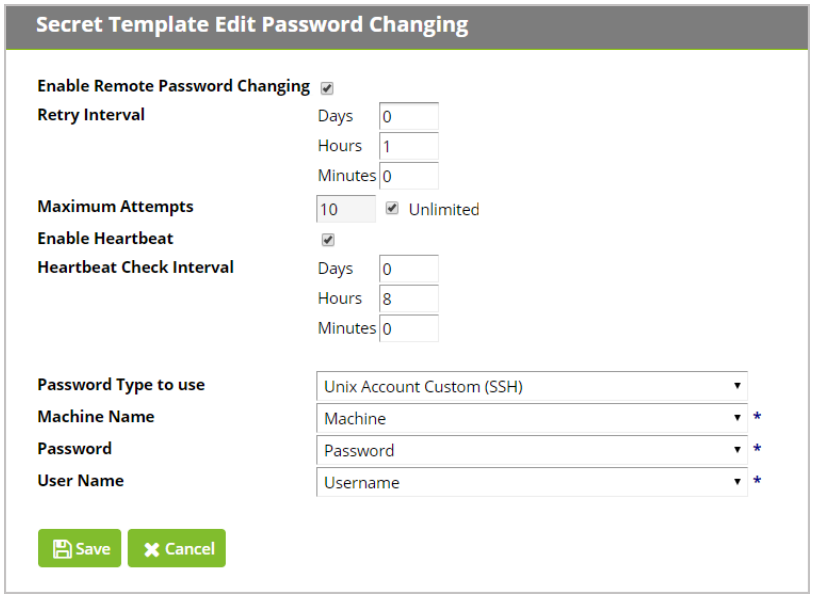
Secret Server checks for expired secrets once every minute. However, if a previous password change attempt failed, Secret Server will not immediately try to change that password again the next time expired secrets are checked. Each secret template has a "Retry Interval" that it uses to determine how often to attempt retries of failed password changes. This exists to prevent unavailable machines or network connection issues from overwhelming the server or network with potentially thousands of password change requests at once. The default retry interval is one hour. To change the default, navigate to **Admin | Secret Templates**, select the template you wish to change from the dropdown menu, and click **Edit**.



At the bottom of the **Secret Template Designer** page, click **Configure Password Changing**



On the **Secret Template Edit Password Changing** page, click **Edit**. Adjust the **Days**, **Hours**, and **Minutes** values of the **Retry Interval**. You can also adjust the **Maximum Attempts** if you want Secret Server to stop attempting to change the password after a specified number of failures. Click **Save** when done.



Configuring Common Criteria

Verify Settings in Government Edition

Verify DPAPI Setting is Enabled

The Delinea Government Edition Installer will automatically set encryption in Secret Server to use DPAPI. To verify that DPAPI is in use, navigate to **Admin > Configuration > Security** tab and ensure that DPAPI is enabled by scrolling to the bottom of the page and checking the existence of the **Decrypt Key to not Use DPAPI** button.

If DPAPI is not enabled in your Secret Server installation, this button will say "**Encrypt Key to Use DPAPI**." To maintain compliance with Common Criteria standards do not decrypt the Secret Server key.

Verify FIPS Mode in Secret Server On-Premises is Enabled

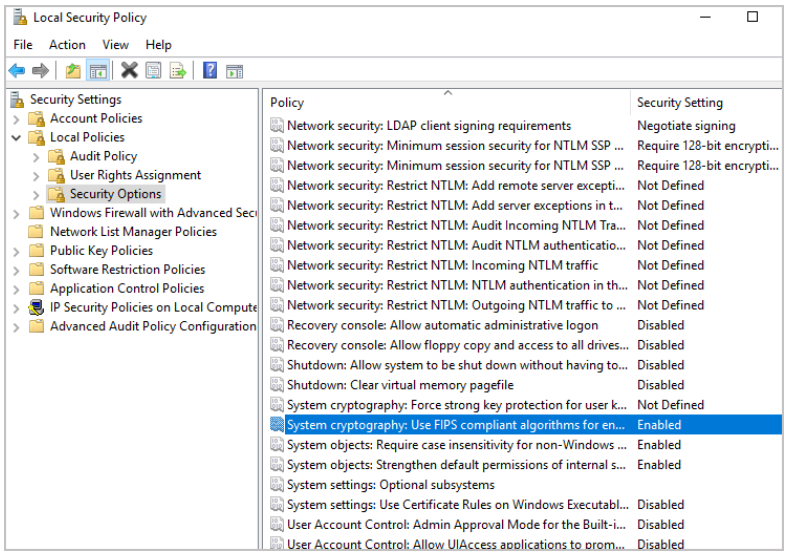
FIPS 140-2 compliance is built-in to Secret Server Cloud and is always on.

Common Criteria evaluations rely on FIPS 140 validations to provide assurance that cryptographic functionality is implemented properly. The Secret Server Government Edition Installer checks for FIPS encryption security standards on the Windows local server and enables FIPS mode by default in Secret Server. To check this setting, go to **Admin > Configuration > Security** tab and ensure that **the Enable FIPS Compliance** line is set to **Yes**.

Manually Enabling FIPS in Secret Server On-Premises on Windows Local Servers

The **Secret Server Government Edition Installer** ensures that FIPS Compliance is automatically enabled in Windows on your local server. To verify or do this manually:

1. Go to **Windows Local Security Policy editor** (secpol.msc)
2. Navigate on your left pane to **Security Settings > Local Policies > Security Options**
3. Find and go to the property of **System Cryptography: Use FIPS Compliant algorithms for encryption, hashing, and signing...**
4. Choose **Enable** and click **OK**.



5. Restart your Server with **iisreset**

The Microsoft .NET implementations of AES and SHA are not FIPS certified so Secret Server uses the Windows API versions for encryption functionality which *are* FIPS certified. **More information on the FIPS certificate numbers for the Windows operating systems, including the algorithm implementations that we use can be found at:**http://technet.microsoft.com/en-us/library/cc750357.aspx

Manually Disabling TLS v1.0

To disable the TLS 1.0 protocol so that IIS does not try to negotiate using the TLS 1.0 protocol, follow these steps:

1. Click **Start**, click **Run**, type regedt32 or type regedit, and then click **OK**.
2. In Registry Editor, locate the following registry key:

   HKey_Local_Machine\System\CurrentControlSet\Control\SecurityProviders \SCHANNEL\Protocols\TLS 1.0\Server

   If the subkeys “Server” or “Client” do not exist, create them under the TLS 1.0 key.
3. On the **Edit** menu for each subkey, click **Add Value**.
4. In the **Data Type** list for each subkey, click **DWORD**.
5. In the **Value Name** box for each subkey, type “**Enabled**,” and then click **OK**.

   *Note If this value is present, double-click the value to edit its current value.
6. Type **00000000** in Binary Editor to set the value of the new key equal to "0".
7. Click **OK**. Restart the Windows Server.

More information can be found at https://support.microsoft.com/en-us/help/187498/how-to-disable-pct-1-0-ssl-2-0-ssl-3-0-or-tls-1-0-in-internet-informat

Microsoft Cryptographic Engine

Secret Server relies on Windows Server 2016 Standard Edition or later to provide protocol and cryptographic functionality. Windows Server 2016 Standard Edition implements a FIPS certified (CMVP #2937) Cryptographic Primitives Library.

More information on the FIPS 140 certificate numbers for the Windows operating systems, including the algorithm implementations that we use can be found at: http://technet.microsoft.com/en-us/library/cc750357.aspx

The use of other cryptographic engines was not evaluated nor supported in the Common Criteria configuration.

Configuring X.509v3 Certificates

In order to maintain compliance with Common Criteria standards, the following installed certificates must follow the standard X.509v3 digital certificate structure.

**Do not use self-signed certificates in production environments.*

Installing Certificate in Trust Store

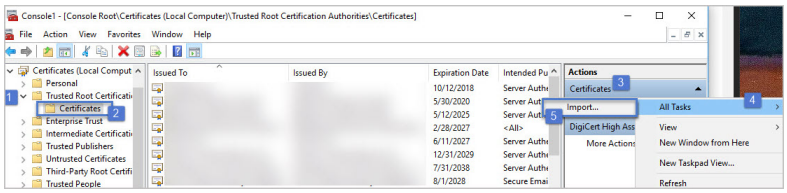
To best protect the users of your Secret Server website, use an SSL certificate from a trusted root authority and install it on your **web server(s)**.

- It is possible to use a self-signed certificate in demo or sandbox environments as long as **all other machines** used to access Secret Serverare also configured to trust the certificate. Otherwise, users will receive warnings about it being invalid and unsafe to navigate to your Secret Server URL.
- Local machines either need to Import your **public certificate** into the **Trusted People** certificate store if you used IIS to create it, or If you used OpenSSL and set up a **custom Root CA** to sign your web server certificate, the Root CA must be imported into the **Trusted Root Certification Authorities** store.
- If a **custom Intermediary CA** is created, this must also be imported into the **Intermediate Certification Authorities** store.

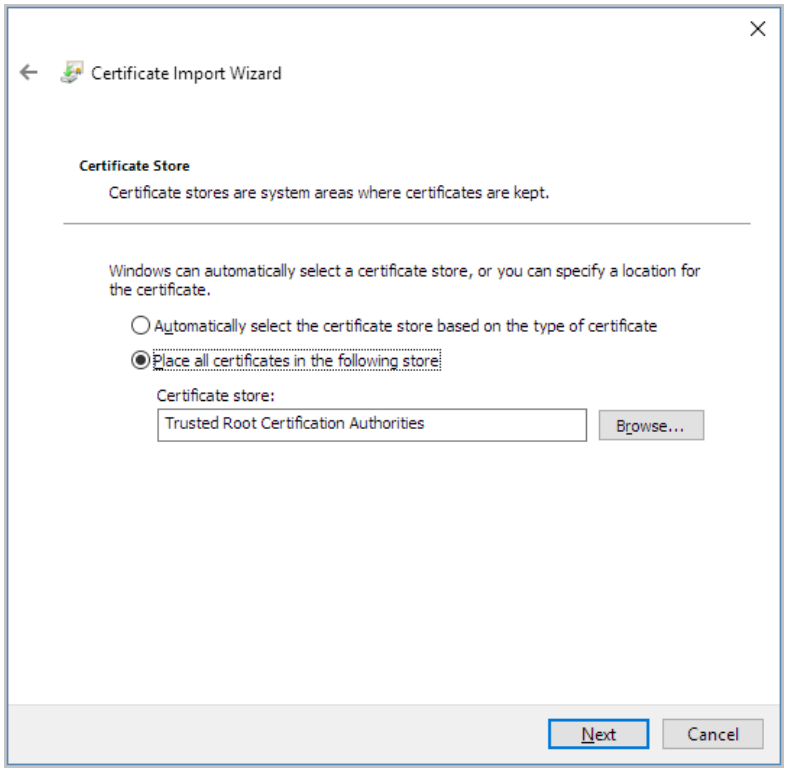
If Secret Server will be communicating with any remote server using TLS, any Root CAs and Intermediate CAs in the trust chains of the certificates presented by those remote servers must be added to the **Trusted Root Certification Authorities** and **Intermediate Certification Authorities** stores respectively. This includes syslog and LDAPS servers.

Follow the steps below to install a certificate in the Trusted Root CA store:

1. On your local machine open **mmc.exe**, select **File**, then **Add/Remove Snap-In…**
2. Next choose **Certificates** and then **Add**. Select **Computer Account**, then **Local Computer**. Click **Finish**.
3. Open **Certificates (Local Computer)**, then expand the **Trusted Root Certification Authorities** store/folder from the left pane and select **Certificates**, then in the right-side **Actions** pane select **More Actions**, **All Tasks**, and **Import…**



4. Click **Next** then select the certificate to import. This is usually a file with a file extension of .cer, .crt, .pem, or .pfx and click **Next**.
5. If prompted, enter the certificate's password and click **Next**.
6. Leave the selected options at the **Certificate Store** step and click **Next**.



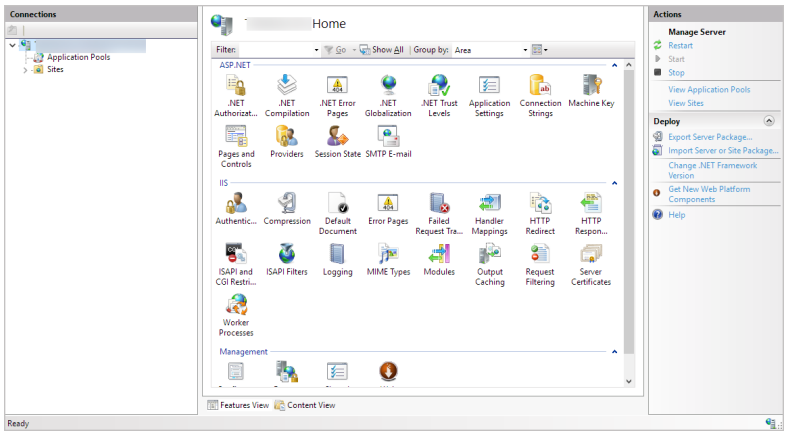
7. Click **Finish** to import the certificate

To import and Intermediate Certificate Authority follow the same steps for a Root Certificate Authority but select **Intermediate Certification Authorities** in step 3 instead of **Trusted Root Certification Authorities**. In step 6, the selected certificate store should also say **Intermediate Certification Authorities**.

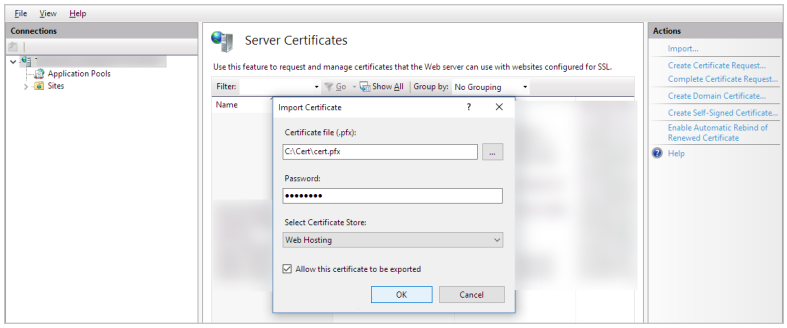
Configuring Web Certificate

To install a web certificate for use by IIS:

1. Open IIS Manager by clicking **Start**, clicking **Run**, typing in "**inetmgr**", and clicking **OK**.
2. Click on the server node (one of the root nodes) in the left panel, and double click on **Server Certificates**.



3. Click **Import...** in the right-hand **Actions** panel



4. Select the certificate to use, enter the password, select “Web Hosting” in **Select Certificate Store**, and click **OK**.

Alternatively, to create a self-signed certificate, click **Create Self-Signed Certificate** on the right panel and type in a Friendly Name.

> Note, using Self-Signed certificates is not compliant with Common Criteria standards. If using a self-signed certificate you must also export and install it on every user machine that connects to the Secret Server website.

Now that your certificate is installed, IIS must be configured to use it.

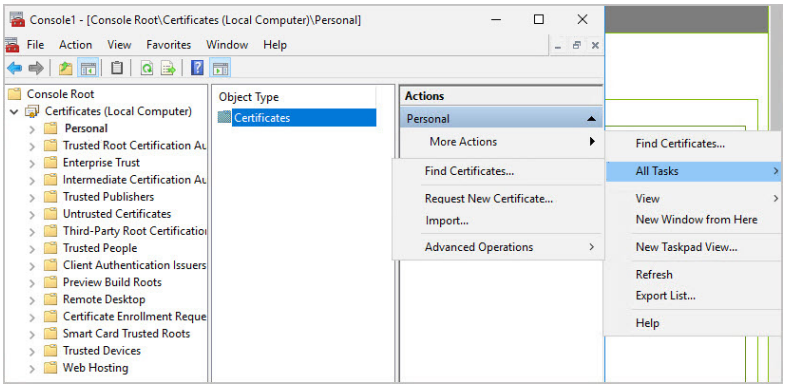
- In IIS Manager, click on your website in the left panel, then click **Bindings** on the right panel.
- If https is not already listed, click **Add**, select **https**, select the correct certificate, and click **OK**.
- If https was already listed, select it, click **Edit**, and select the correct certificate.

Configuring Client Certificates

Client Certificates can be used for **LDAPS** connections and for **Secure TCP** Syslog connections if your LDAPS or Secure TCP Syslog server requires one to be used.

You will need to install a Client Certificate on your **web server(s)** in the Local Computer’s **Web Hosting** (used by newer versions of IIS) or **Personal** Stores:

1. On your local machine open **mmc.exe**, select **File**, then **Add/Remove Snap-In…**
2. Next choose **Certificates** and then **Add**. Select **Computer Account**, then **Local Computer**. Click **Finish**.
3. Open **Certificates (Local Computer)**, then import your certificate by selecting the **Personal** or **Web Hosting** store/folder from the left pane, then **All Tasks**, and **Import…**

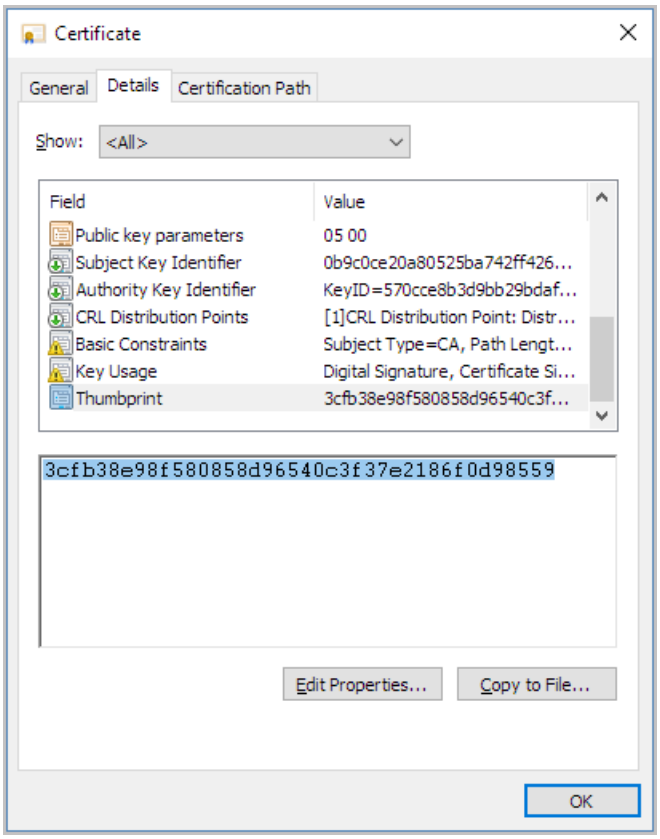


4. Follow the Import Wizard to complete the installation of the client certificate.If you are using an older version of IIS, or IIS is not installed on your machine, the Web Hosting store will not exist. In that case, import into the Personal store.

To enable Secret Server to send the client certificate to remote servers, you must first get the certificate thumbprint. This thumbprint will be added to Secret Server and used to identify which certificate to send when a client certificate is requested.

To get the certificate thumbprint:

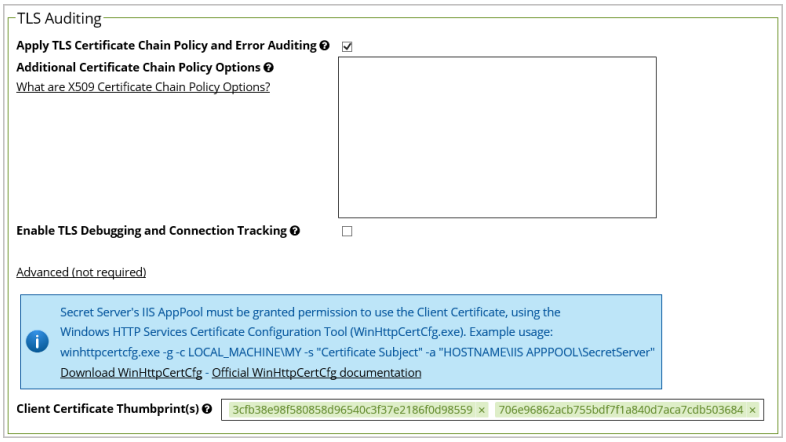
1. Right-click on the certificate file in Windows Explorer or the certificate listed in MMC and select **Open**.
2. Select the **Details** tab.
3. Scroll down in the list of fields until you find **Thumbprint**. Select it. The value will be displayed in the box below the field list.



4. Select the text in this box and copy it by pressing **Ctrl+V**.

Now that you have the thumbprint copied to the clipboard, you can add it to Secret Server:

1. In Secret Server, select **ADMIN > Configuration**.
2. Select the **Security** tab and click **Edit**.
3. Find the **TLS Auditing** section.



4. Check **Apply TLS Certificate Chain Policy and Error Auditing**.

5. Click **Advanced (not required)** to expand that section.
6. Paste the thumbprint in the **Client Certificate Thumbprint(s)** field. If you need to use more than one client certificate (for example, if you havemultiple Secret Server nodes with different client certificates), you can paste multiple thumbprints in this field.
7. Click **Save** when done.

Certificate Identifiers Overview

Certificates used for server authentication are verified by Secret Server using the verification provided by Microsoft Windows' Schannel. A certificate must contain a reference identifier that the client can use to verify that the server is presenting a matching certificate. This identifier can be contained in either the certificate **Subject** or **Subject Alternative Name (SAN)**. In order for the Subject to be verified as valid for a given server, it must contain a CN entry that contains the **Fully-Qualified Domain Name (FQDN)** of the server. If a certificate needs to be valid for more than one reference identifier, the certificate can contain a SAN that contains multiple entries. These entries can include both DNS names and IP addresses. If a certificate contains a SAN, the CN in the subject is ignored and only the entries in the SAN are used to validate the certificate against the server.

Configuring Transport Layer Security (TLS)

Common Criteria users must configure TLS in Secret Server. To enable TLS, follow the steps in the below sections:

TLS Configuration with AD

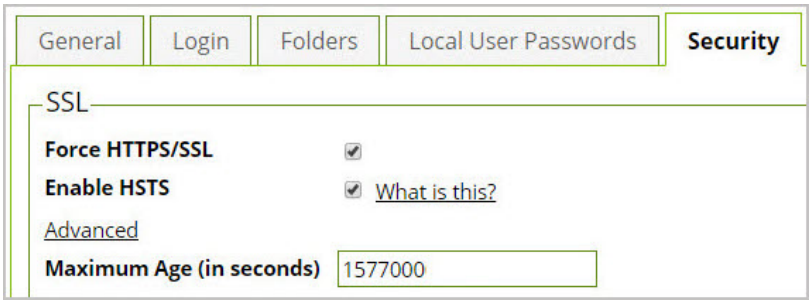
To configure TLS with Active Directory, follow the steps in section **7.1.3 Configuring TLS with Active Directory.**

TLS Configuration with Syslog

To configure TLS with Syslog, follow the steps in section **13.3 Configuring Auditing for TLS Connections**.

TLS Configuration with IIS

According to Common Criteria security standards, you must force Secret Server through HTTPS/SSL for all website connections, ensure you are using an https URL in your Secret Server browser, then navigate to **Admin > Configuration > Security** tab, click **Edit** and then check the **Force HTTPS/SSL** checkbox, followed by the **Enable HSTS**. We recommend adjusting the Maximum Age (in seconds) to a high number, the example below allows for approximately six months, but we recommend up to one year. **Save** changes.



For more information on HSTS, see "Securing Traffic with HTTP Strict Transport Security" on page 785

Local Auditing

Understanding Local Audit Records and Reports

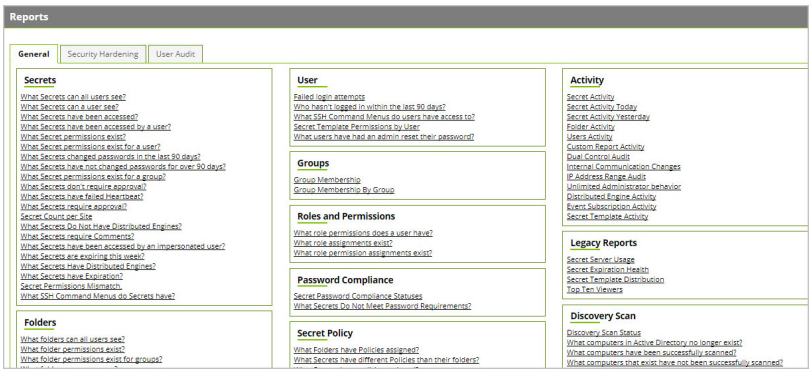
By design, Secret Server locally audits all actions taken within Secret Server. Secret Server's auditing capacity is not configurable. See **Auditable Events** for details on Secret Server items that are audited and corresponding user permissions required for accessing audit records.

Various User Permissions are tailored for specific kinds of audits, listed below. To adjust user permissions, go to **Admin | Roles** and create a new role or click an existing role to edit the permissions on it. You can assign roles to individual users by going to **Admin | Users** and assigning roles.

- **Add Secret Custom Audit**
  Allows a user to make a custom audit entry when accessing a Secret using the web services API.
- **User Audit Expire Secrets**
  Allows a user to view the User Audit report, which shows all secrets that have been accessed by a particular user in a specified date range. Also allows the user to force expiration on all these secrets, which would make Secret Server automatically change the password.
- **View Configuration Unlimited Admin**
  Formerly "View Unlimited Admin Configuration," allows a user to view theUnlimited Admin Mode configuration. Also allows a user to view the Unlimited Admin Mode audit log.
- **View Secret Audit**
  Allows a user to view Secret Audit.
- **View User Audit Report**
  Allows a user to view, but not edit, the User Audit Report.

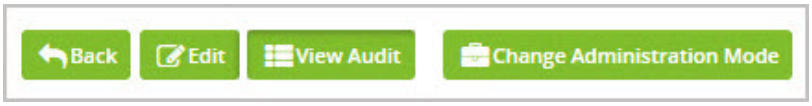
Accessing Local Reports

To view a list of out of the box reports, navigate to the **Reports** tab from an Administrator account.



For customized reports and creation, see the options available to you at the bottom right corner of the Reports page. The **View Audit** button on the bottom left corner will show you a local audit for any reports that have been viewed.

Administrator actions have audit trails inside Secret Server, these are generally found as a **View Audit** button on any landing page under the **Admin** tab. For example, at the bottom of **Admin | Configuration** you can view the audit for every edit that has occurred on the **Configuration** page:



For User Audits, navigate to **Reports | User Audit** tab, then select a user and time period to view the audit of all actions taken by that user in Secret Server.

Configuring Local Windows Event Log Auditing

Outside of Secret Server, you can also send audit logs as EVT records to your Windows Server's Windows Event Log locally**.**

Note that the Windows Event Log is tied to the syslog implementation, because of this, disabling syslog will disable Windows Event Logs as well. If Syslog/CEF Logging is not yet enabled within Secret Server, go to **Admin | Configuration** and **Edit** the General tab to check **Enable Syslog/CEF Logging** followed by **Write Syslogs As Windows Events**. *Syslog/CEF Logging configuration also detailed in section 13.3 Configuring Auditing for TLS Connections*

The following steps walk you through how to grant required permissions to the Application Pool outside of Secret Server to successfully write audits to the Windows Event Log:

Granting Application Pool Access to Windows Event Log

When the database becomes inaccessible, Secret Server will try to log errors to the Windows Event Log. By default, however, Network Service and standard service accounts will not have permissions to the Event Log. *Permissions must be manually added to the Application Pool for specific Event Log registry keys*.

> Changes made to the Windows registry happen immediately, no ***backup is created automatically. Do not edit the Windows registry unless youare confident about doing so. Using Registry Editor incorrectly can causeserious, system-wide problems that may require you to re-install Windows tocorrect them. Microsoft cannot guarantee that any problems resulting from*** the use of Registry Editor can be solved. Use this tool at your own risk.

Required Registry Permissions

The following permissions are required for the Identity configured on the Secret Server Application Pool in IIS (Network Service, IIS APPPOOL\SecretServer, etc.).

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog

Applies to "This key and subkeys".

- Read
  - Query Value
  - Enumerate Subkeys
  - Notify
  - Read Control
- Set Value
- Create Subkey

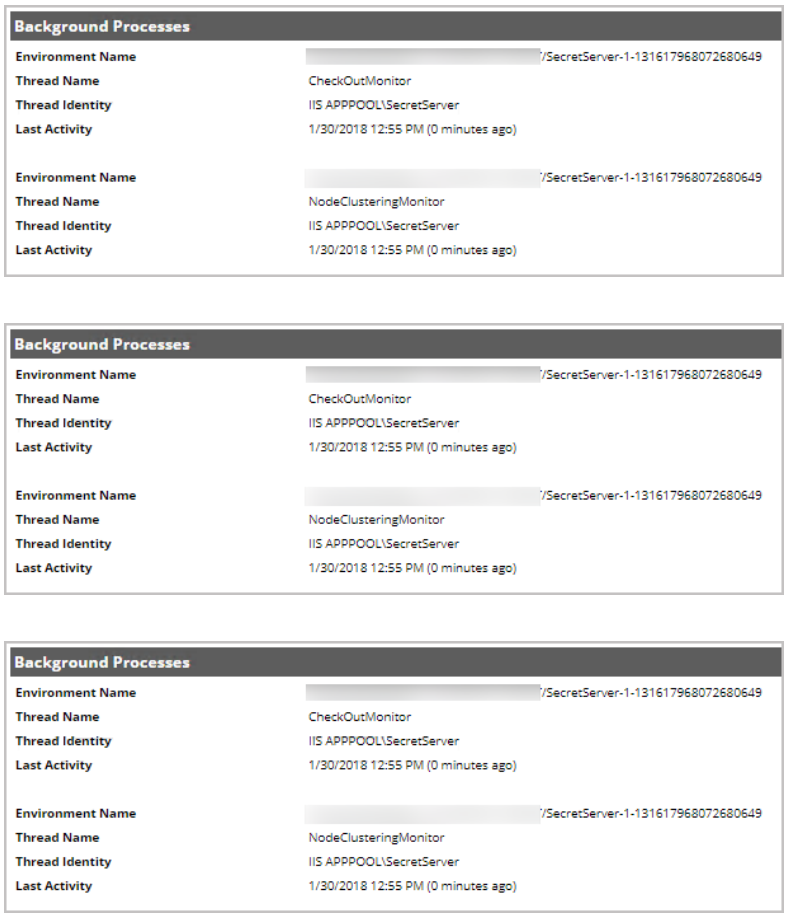
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Security

Applies to "This key and subkeys".

- Read
  - Query Value
  - Enumerate Subkeys
  - Notify
  - Read Control

How to Apply Windows Event Log Permissions

1. Determine the account that is running Secret Server. This can be done by Secret Server. logging in to Secret Server, **Admin | Diagnostics**. Click the button to**Show Background Processes**. Look for any of the "Thread Identity" labels.These will contain the identity of Secret Server (often NT AUTHORITY\NETWORK SERVICE or IIS APPPOOL\SecretServer).



Background Processes

Environment Name	[redacted]/SecretServer-1-131617968072680649
Thread Name	CheckOutMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)
Environment Name	[redacted]/SecretServer-1-131617968072680649
Thread Name	NodeClusteringMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)

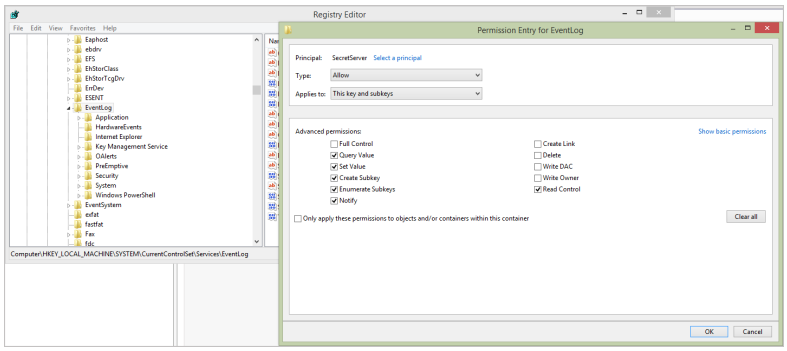
Background Processes

Environment Name	[redacted]/SecretServer-1-131617968072680649
Thread Name	CheckOutMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)
Environment Name	[redacted]/SecretServer-1-131617968072680649
Thread Name	NodeClusteringMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)

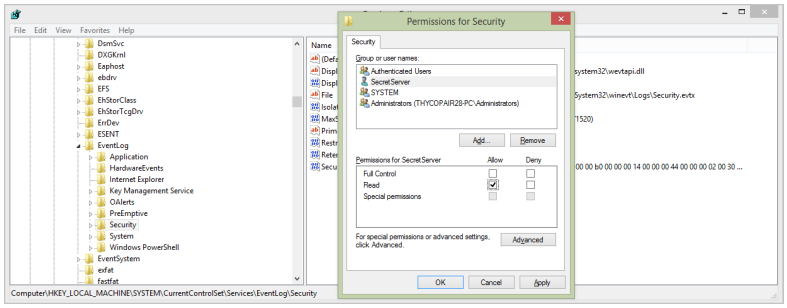
Background Processes

Environment Name	[redacted]/SecretServer-1-131617968072680649
Thread Name	CheckOutMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)
Environment Name	[redacted]/SecretServer-1-131617968072680649
Thread Name	NodeClusteringMonitor
Thread Identity	IIS APPPOOL\SecretServer
Last Activity	1/30/2018 12:55 PM (0 minutes ago)

2. Open the **Registry Editor** on the machine running Secret Server (**Start | run-regedit**, depending on OS)
3. On the left, navigate to **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog**
4. Right click on the **EventLog** folder in your registry editor and select **Permissions**, then click the **Advanced** button
5. On the Permissions tab, Click **Add**
6. Click **Select a principal**.
7. Enter the name of your app pool's account (see step 1, for example: "IIS APPPOOL\SecretServer") in the box listed under **Enter the object name to select (examples).** Click **Check Names**, Once "SecretServer" (or the name of your Secret Server app pool) is listed in the box and underlined, click **OK**
8. Under **Basic permissions:**, check **Read**.
9. Click **Show advanced permissions**, then check **Set Value** and **Create Subkey** under **Advanced permissions:**.



10. Click **OK** on the remaining dialogs and **Apply** the permissions. You should be back on the main Registry Editor window.
11. Navigate to **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Security**, right-click and select **Permissions...**
12. Click **Add...**, find the account running Secret Server (like in step 7), then click **OK**.
13. Check **Read** in the **Allow** column, then click **OK** to apply the permission.

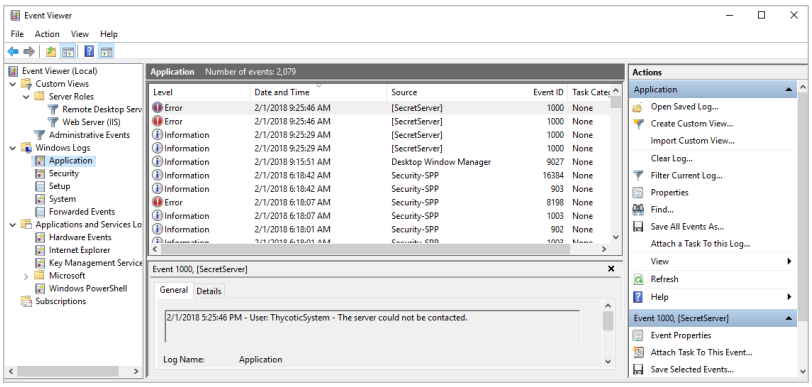


14. In Secret Server, navigate to **Admin > Configuration > General** tab to verify that the Application Setting **Write Syslogs as Windows Events** is set to **Yes**.

Accessing Windows Event Logs

After setting up Windows Event Logs to consume Secret Server logs you can access logs for local auditing and troubleshooting purposes through the **Windows Event Log Event Viewer** on Secret Server's local server.

To find the Windows Event Log, open the Windows **Event Viewer** from the local server and navigate to **Windows Logs | Application**. Event Logs for Secret Server will be listed as **[SecretServer]** under the **Source** column:



External Auditing

Security—Connecting to an External Audit Server

Using the **Secure TCP** Syslog/CEF Protocol will make Secret Server try to establish a secure connection to your external audit server. Secret Server will only use TLS v1.2 or v1.1 for security reasons.

If your audit server requires Client Certificates, see **Configuring Client Certificates**. If configured, Secret Server will pass the Secret Server certificate to the external audit server when it connects, so the audit server can validate the Secret Server machine.

If the connection between Secret Server and the external audit server is disconnected or cannot be established, Secret Server will log an internal error and automatically try to re-send any missed audit messages later. For more information, see **Determining Status of a Remote Audit Server**.

For a connection to be successfully established using Secure TCP protocol, Secret Server must trust the SSL certificates being used by the audit server. If the audit server is using a self-signed certificate, the Certificate Authorities that created it must be installed on the Secret Server machine. If Secret Server has issues trusting the certificate, details will be logged internally.

More information on creating your own Certificate Authorities can be found at the OpenSSL Certificate Authority.

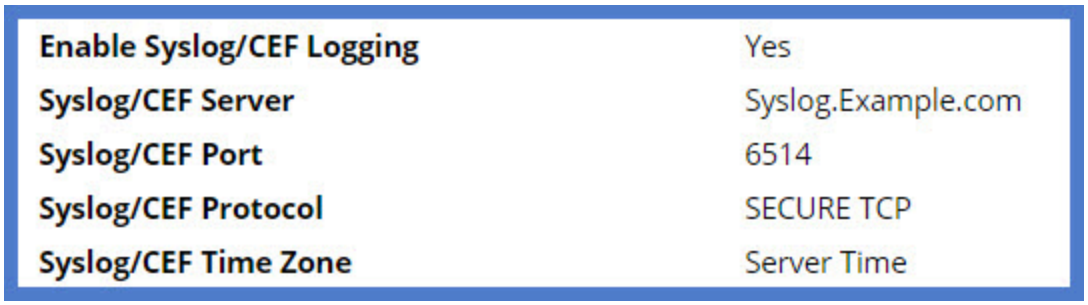
Configuring Syslog/CEF External Audit Server

Compatible Audit Servers

- syslog-ng
- Any Audit server that accepts TLS encrypted messages using the BSD Syslog Protocol

Configuring Your External Audit Server

1. Navigate to **Admin | Configuration**, then click **Edit**.
2. You will see a configuration area in the **General tab** under **Application Settings**to **Enable Syslog/CEF Logging**. Check this box to enable:



Syslog/CEF may require an additional license key. To install licenses, navigate to **Admin | Licenses** and select **Install New License**. Once installed, you will need to activate your license. Contact your Delinea Sales Representative if you have questions about your licensing.

- **Syslog/CEF Server.** Configure the Syslog/CEF IP address for the IIS Server hosting the Syslog/CEF web application.
- **Syslog/CEF Port.** Next, configure the port number where the logging information will be passed for the **Syslog/CEF Port.** 6514 is the default port for Secure TCP Syslog. Secret Server requires outbound access to this server and port so communication can pass freely.
- **Syslog/CEF Protocol**. Set the Syslog/CEF Protocol to **Secure TCP**. This setting will accept either TLS v1.2 or v1.1 for added security, because other versions of SSL (i.e. SSL v3 and TLS v 1.0) have known weaknesses.
- **Syslog/CEF Time Zone**. Lastly set Syslog/CEF Time Zone to **UTC Time** or **Server Time** depending on your preference.

The standard for Syslog is ISO timestamps; however, some still use the legacy format. Syslog is the default for upgrades to allow current configurations to retain their behavior, and ISO format is the default in new instances. Syslog format: Jun 23 2022 11:22:33. ISO 8601 format: 2022-06-23T11:22:33.000. You must enable the configuration preview to modify this setting.

Caching Syslog Audits

Once secure logging is enabled in Secret Server, if the connection breaks between the external Syslog server and Secret Server, failed syslog messages will be cached in the Secret Server database and re-sent at regular intervals until the connection between the syslog server and Secret Server is reestablished.

Compatibility Notes Related to Using Client Certificates

If you are using a Client Certificate, Secret Server's IIS AppPool must be granted access to use the certificate using the Windows HTTP Services Certificate Configuration Tool (WinHttpCertCfg.exe). Otherwise if Secret Server is configured to use a Client Certificate, and IIS doesn't have permission, you will see errors like this in the logs:

TLS Error Detected (Authentication Error connecting to IP:PORT) - The credentials supplied to the package were not recognized.

Example usage:
cd C:\Program Files (x86)\Windows Resource Kits\Toolswinhttpcertcfg.exe -g -c LOCAL_MACHINE\MY -s "CertificateSubject" -a "HOSTNAME\IIS APPPOOL\SecretServer"

Download the Windows HTTP Services Certificate Configuration Tool (WinHttpCertCfg.exe).

If you are using a Client Certificate and a Syslog-NG logging server, you may occasionally see this message in the main Syslog-NG log file:

SSL error while reading stream; tls_error='SSL routines:ssl_get_prev_session:session id context uninitialized'

On the Secret Server side, it shows up like this:

TLS Error Detected (Authentication Error connecting to IP:PORT) - Authentication failed because the remote party has closed the transport stream.

This error is caused because Windows tries to cache secure connections when client certificates are in use, but because Syslog-NG has not configured their OpenSSL “session id context”, OpenSSL gives this error when it tries to resume a previous session.

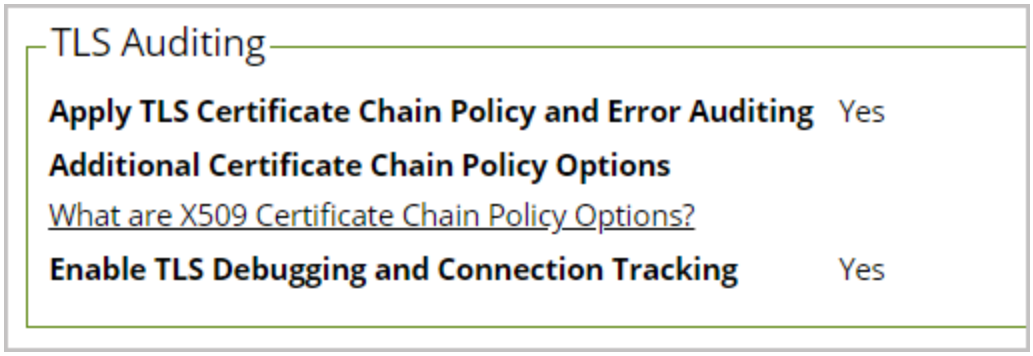
Secret Server will automatically reconnect and resend any missed messages, so these errors really should cause no impact. However if you would like, you can disable Window’s secure connection caching, by adding the ClientCacheTime setting set to 0 (zero) in the Registry, and then doing a reboot. This did not cause any significant performance impact in our internal testing, but your mileage may vary.

If Syslog-NG configures their OpenSSL session ID context, this error message correction will no longer be needed.

Configuring Auditing for TLS Connections

To track problems with TLS connections including whenever the connection might fail, enable the TLS Certificate Chain Policy and Error Auditing in Secret Server by navigating to **Admin > Configuration > Security** tab, then scrolling down to the TLS Auditing section.

- Ensure that the **Apply TLS Certificate Chain Policy and Error Auditing** is set to **Yes.** If not set to yes, Client Certificates cannot be used.
- Ensure that the **Enable TLS Debugging and Connection Tracking** is set to **Yes**. When set to yes, Secret Server will send information logs to your audit server about when TLS connections are opened or closed. If debug logging is enabled in "web-log4net.config" detailed information about X509 certificate chain validation will also be logged. Note that this setting may create a lot of extra messages in your log files.



To Edit click the **Edit** button and **check** the setting you want to turn on or uncheck the setting you want to turn off.

If the TLS connection breaks, an error message will be logged in the local audit trail and Secret Server will keep trying every 60 seconds to reestablish the TLS connection to the syslog server.

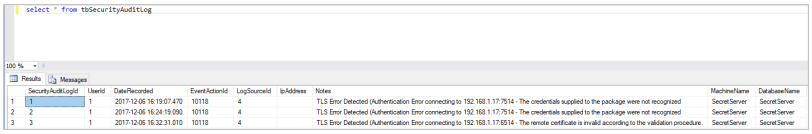
If Secure TCP is used for the Syslog/CEF Protocol, and there are one or more Client Certificate Thumbprints entered, Secret Server will check the Local Computer's Web Hosting and Personal certificate store and use the first one it finds. For more information see **Configuring Client Certificates**.

To add Client Certificate Thumbprints, you can copy and paste a list in bulk after clicking Edit, then Advanced (not required). Separate each SSL Certificate SHA1 Thumbprint (40 characters each) with a semicolon (up to ten total are allowed).

Determining Status of a Remote Audit Server

To view the logs for any TLS-Connection related errors:

- Open **Microsoft SQL Server Management Studio**, navigate to your SecretServer database (**DBMachineName > Databases > SecretServer**) and run a **New Query**.
- Type **select * from tbSecurityAuditLog** to view the events from your TLS Audit:



For more detailed troubleshooting reporting, refer to the logs in File Explorer on Secret Server's web server C:\inetpub\wwwroot\SecretServer\log) including ss.log, ss-BSSR.log (Background Scheduler), ss-BSWR.log (Background Worker) for any errors that might crop up.

Management Functions

This section describes management activities and corresponding roles of the evaluated security functionality

Roles and Management Functions

Role	Management Functions
Read-only User	Search and list Secrets
User	Use Secret/Launch session
User	Request access to Secret
Administrator	Create, view, expire, edit, and assign Secrets
Administrator	Perform bulk operations on Secrets
Administrator	Create and manage groups
Administrator	Create and manage roles, assign roles to users

Role	Management Functions
Administrator	Create and manage containers (folders)
Administrator	Create and manage Secret policy
Administrator	Configure TOE SF (see Table 16)
Administrator	Create, manage, and unlock local accounts
Administrator	Configure IIS, SQL, syslog
Administrator	Update TOE

By default each User Role is attached to various Permission Sets. To view the specific permissions that each role is attached to, navigate to **Admin | Roles** and click into the user roles listed to see the list of permissions.

Organizations can tailor these user roles to maintain whatever permissions settings are required for your specific user environment.

The following table specifies the specific user roles required to allow each management activity listed.

Requirement	Management Activities	Role
ESM_EAU.2	Management of authentication data for both interactive users and authorized IT entities (if managed by the TSF)	Administrator
ESM_EID.2	Management of authentication data for both interactive users and authorized IT entities (if managed by the TSF)	Administrator
ESM_ICD.1	Definition of identity and credential data that can be associated with users (activate, suspend, revoke credential, etc.)	Administrator
	Management of credential status	Administrator
	Enrollment of users into repository	Administrator
ESM_ICT.1	Configuration of circumstances in which transmission of identity and credential data (and object attributes, if applicable) is performed	Administrator
FAU_STG_ EXT.1	Configuration of external audit storage location	Administrator
FIA_AFL.1	Management of the threshold for unsuccessful authentication attempts	Administrator
	Management of actions to be taken in the event of an authentication failure	Administrator

Requirement	Management Activities	Role
FIA_USB.1	Definition of default subject security attributes, modification of subject security attributes	Administrator
FMT_MOF.1	Management of sets of users that can interact with security functions	Administrator
FMT_SMR.1	Management of the users that belong to a particular role	Administrator
FTA_SSL.3	Configuration of the inactivity period for session termination	Administrator
FTA_TAB.1	Maintenance of the banner	Administrator
FTP_ITC.1	Configuration of actions that require trusted channel (if applicable)	Administrator
FTP_TRP.1	Configuration of actions that require trusted path (if applicable)	Administrator

Distributed Engine Hardening

Introduction

This topic discusses best practices for hardening Secret Server distributed engine servers.

If attackers compromise one of the DE servers, they would have access to all critical DBs, applications, and network devices at the network level. DEs do not store any passwords, PII, or user data in any configuration files.

Due to their intrinsic nature, some PowerShell script run by DEs may expose API usernames or passwords in the PowerShell log.

General Hardening Steps

Restrict RDP Connections

- Limit RDP connections to all PAM Server, except for PAM admins and some users from the hosting team.
- If there is no firewall segmentation in LAN network, you can accomplish this with the Windows OS firewall.

Restrict Incoming Port Access to All DE Servers

- Allow only RDP port access from some internal IPs.
- Allow a SSH proxy port coming from the user's LAN.
- Block all other incoming ports.

Remove Unnecessary User Groups

For administrator and Remote Desktop user groups:

- Remove default domain admins, administrator and some common groups.
- Create one group that is going to have access these servers.

- Disable the built-in local administrator user.

Rename Default Accounts

- Change the names of both the administrator and guest accounts to names that do not indicate their permissions.
- Create a new locked and unprivileged "administrator" user name as bait.

Disable Services

Disable these services:

- Routing and remote access
- Smart card
- Smart card removal policy
- SNMP trap
- Special administration console helper
- Windows error reporting service
- WinHTTP Web proxy auto-discovery service

Restrict Network Protocols

Keep these:

- Client for Microsoft network
- File and printer sharing for Microsoft network
- Internet protocol version 4 (TCP/IPv4)

Remove these:

- QoS packet scheduler
- Link-layer topology discovery mapper IO driver
- Link-layer topology discovery responder

Validate Server Roles

Ensure only the minimum roles and features that are required are defined on the DE Servers. Remove all unnecessary roles and features.

The following roles are removal candidates, not ones to keep.

Roles

Application Server

- TCP port sharing
- Windows process activation service support
- Named pipe activation
- TCP activation

Remote Access

- Direct access and VPN (RAS)
- Routing
- Web application proxy (with dependent features)

Web Server (IIS)

- Web server
- Health and diagnostic
- Logging tools
- Tracing

Security

- Centralized SSL certificate support
- Client certificate mapping authentication
- Digest authentication
- IIS client certificate mapping authentication
- IP and domain restrictions
- URL authentication

Application Development

- Server side includes
- Web socket protocols
- Windows deployment services (with dependent features), including all child roles

Features

- Group policy management
- IIS hostable Web core
- Ink and handwriting services

- Media foundation
- RAS connection manager administration kit (CMAK)
- Remote server administration tools, including all child features.
- Windows internal database
- SMB 1.0/CIFS file sharing support

SSL/TLS Settings

Keep your server SSL/TLS settings up to date. Among other settings, the different protocols and cipher suites can be vulnerable to different attacks on SSL/TLS.

- Disable SSL 2.0
- Disable SSL 3.0
- Disable TLS 1.0
- Disable TLS 1.1
- Enable TLS 1.2

GPO Hardening

The following are recommended settings for Microsoft Group Policy Objects (GPO).

User Configuration > Policies > Administrative Templates > Control Panel/Personalization

Vulnerability:

There is no protection against a user with physical and remote desktop access to the server.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Enable screen saver	Enabled
Force specific screen saver	Enabled
Password protect the screen saver	Enabled
Screen saver timeout	Enabled Seconds: 600

Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies /Security Options

This setting enables advanced auditing in the operating system.

Policy	Recommended Value
Audit: Force audit policy subcategory settings to override audit policy category settings	Enabled

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Logon Account

Vulnerability:

Lack of information on unauthorized user login attempt. Lack of this type of information prevents identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Credential Validation	Success, Fail
Other Account Logon Event	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Account Management

Vulnerability:

Lack of information on user management in the system (addition and removal of users). Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Application Group Management	Success, Fail
Computer Account Management	Success, Fail
Distribution Group Management	Success, Fail
Other Account Management Events	Success, Fail

Policy	Recommended Value
Security Group Management	Success, Fail
User Account Management	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Logon\Logoff

Vulnerability:

Lack of information on unauthorized user login attempt. Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Account Lockout	Success, Fail
Logoff	Success, Fail
Logon	Success, Fail
Network Policy Server	Success, Fail
Other Logon\Logoff Event	Success, Fail
Special Logon	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Object Access

Vulnerability:

Lack of information on access to sensitive files and folders in the system. Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

Applying Auditing for Success can overload the system. In case an overload is created, it is recommended to apply the auditing for Failure only.

Policy	Recommended Value
Application Generated	Success, Fail
Certification Services	Success, Fail
Detailed File Share	Fail
File Share	Success, Fail
File System	Success, Fail
Kernel Object	Success, Fail
Registry	Success, Fail
Removable Storage	Success
SAM	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Policy Change

Vulnerability:

Lack of information on changes in the policy. Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Audit Policy Change	Success, Fail
Authentication Policy Change	Success, Fail
Authorization Policy Change	Success, Fail
Filtering Platform Policy Change	Success, Fail
MPSSVC Rule-Level Policy Change	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > Privilege Use

Vulnerability:

Lack of information on the use of system authorizations. Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Non Sensitive Privilege Use	Success, Fail
Sensitive Privilege Use	Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Advance Audit Policy Configuration > System

Vulnerability:

Lack of information on system start-up, shutdown and system changes. Lack of this type of information will prevent identification of intruders to the system, as well as inability to check access to the server or its resources.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Other System Events	Success, Fail
Security State Change	Success, Fail
Security System Extension	Success, Fail
System Integrity	Success, Fail

Computer Configuration > Policies > Windows Settings > Security Settings > Event Log

Vulnerability:

There is a risk that many log records will not be saved due to the file's size.

Severity of the damage:

Medium

Operational aspects:

None

Policy	Recommended Value
Maximum application log size	100032 KB
Maximum security log size	100032 KB
Maximum system log size	100032 KB
Prevent local guests group from accessing application log	Enabled
Prevent local guests group from accessing security log	Enabled
Prevent local guests group from accessing system log	Enabled
Retention method for application log	As needed
Retention method for security log	As needed
Retention method for system log	As needed

Computer Configuration > Policies > Windows Settings > Security Settings > Registry

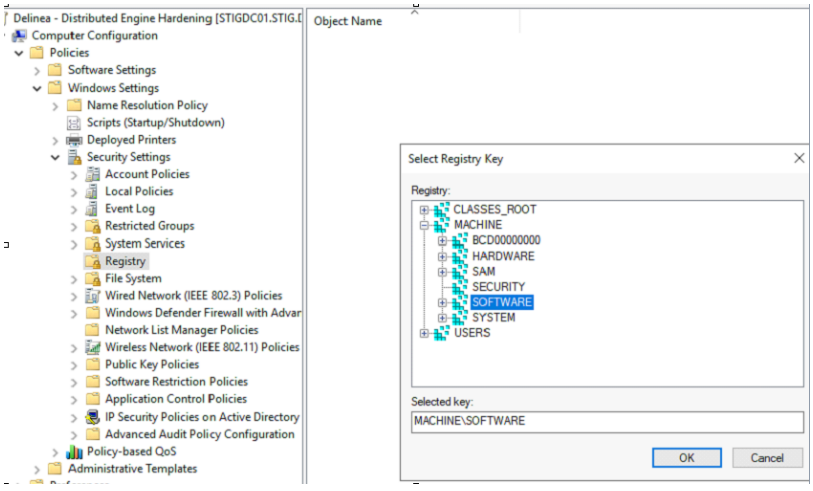
The purpose of this GPO setting is to add auditing to the following registry keys:

- HKLM\SYSTEM
- HKLM\SOFTWARE

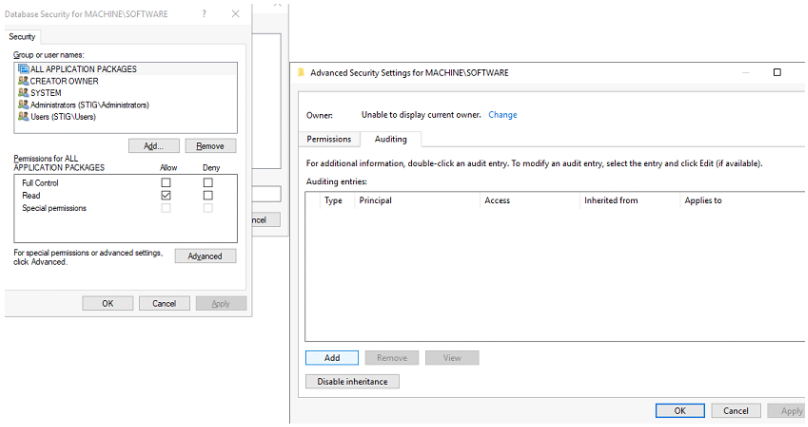
Auditing should be applied according to the following parameters:

- Audit - Success only: Set Value
- Audit - All: Create Subkey, Create Link, Delete, Read Permissions, Change Permissions

1. Right click on Registry, select Add Key, then select MACHINE\SOFTWARE.

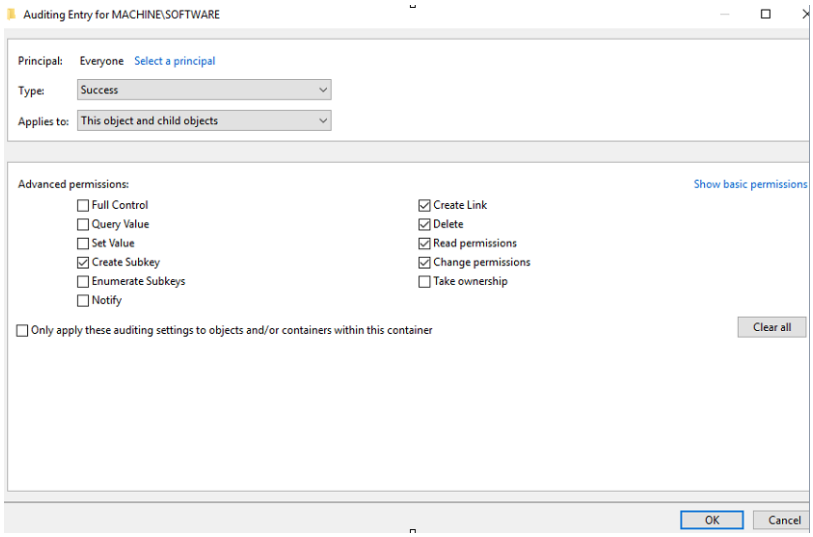


2. Click Advanced, select Auditing tab, click Add.

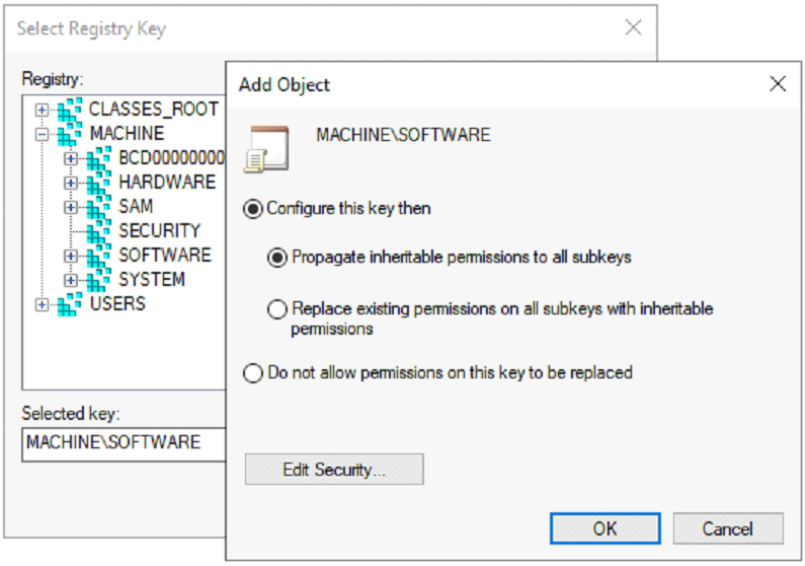


3. Change Principal to Everyone, select Show Advanced Permissions, select the following boxes:

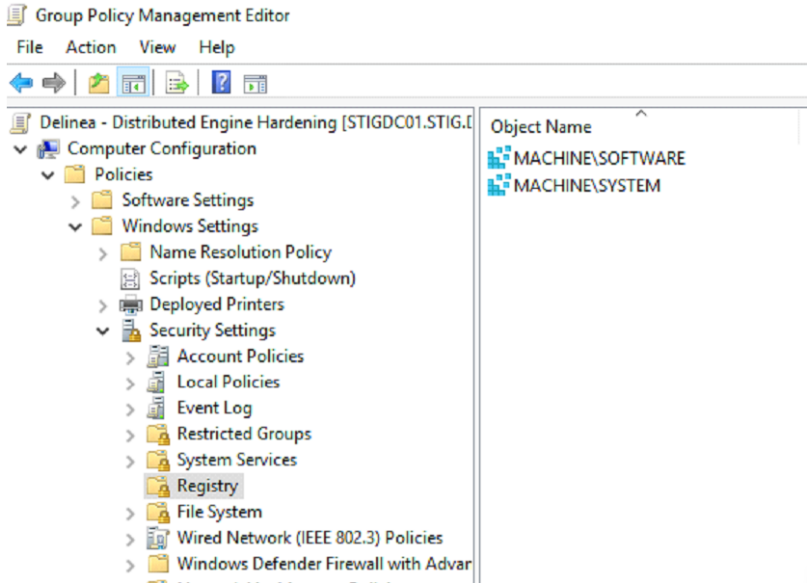
- Create Subkey
- Create Link
- Delete
- Read Permissions
- Change Permissions



4. Click OK, then click Apply Settings.



5. Perform the same steps above for MACHINE\SYSTEM.



Computer Configuration > Policies > Windows Settings > Security Settings > File System

The purpose of this GPO setting is to add auditing to the following directories:

- `%SystemRoot%\System32\Config`
- `%SystemRoot%\System32\Config \RegBack`

Vulnerability:

Lack of information on delete, change of authorizations, gain ownership of sensitive files, or any attempt to do so, will prevent the ability to identify unauthorized access and therefore will make it difficult to prevent such attempts.

Severity of the damage:

Medium

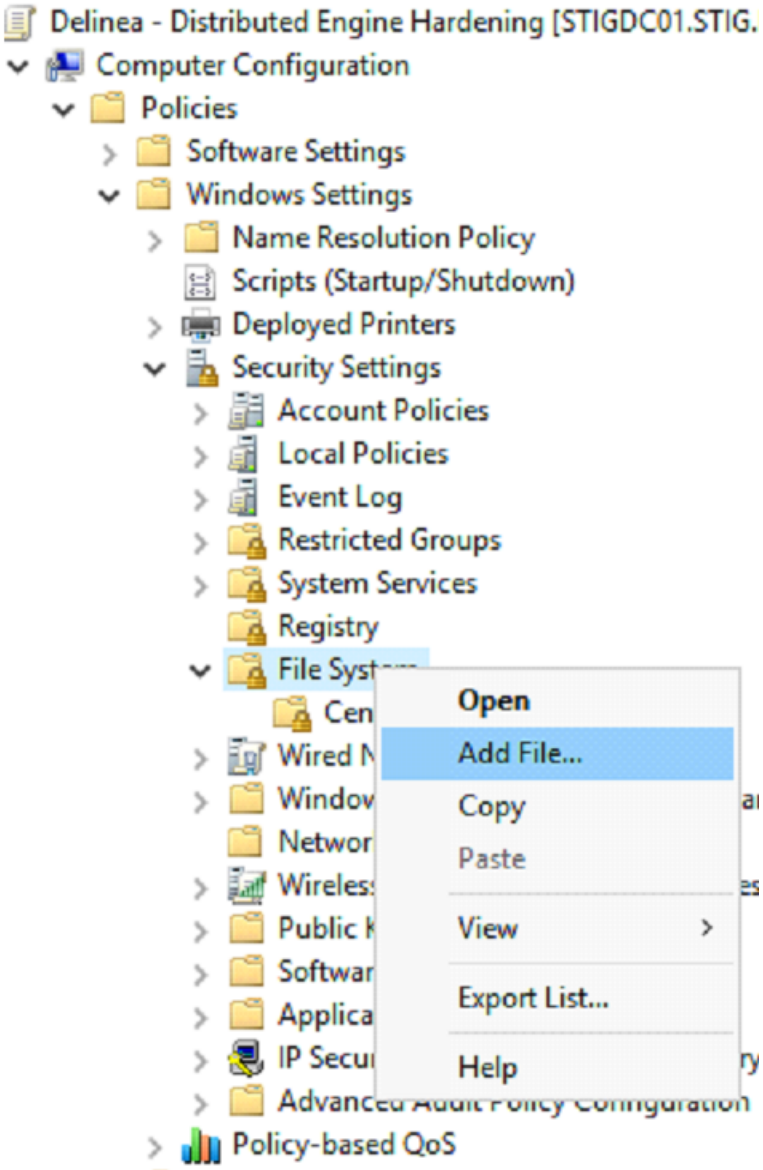
Operational aspects:

None

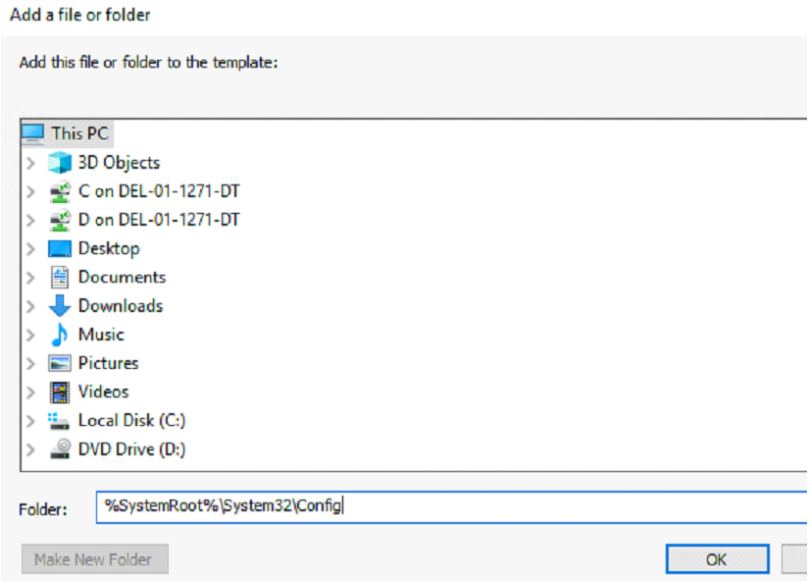
Permissions and auditing should be applied according to the following parameters:

- Audit-Failure only: Traverse Folder\ Execute File, List Folder\ Read Data, Read Attributes, Read Extended Attribute.
- Audit - All: Create Files\ Write Data, Create Folders\ Append Data, Write Attributes, Write Extended Attributes, Delete Subfolders And Files, Delete, Change Permissions, Take Ownership.
- Permissions: Administrator, System - Full

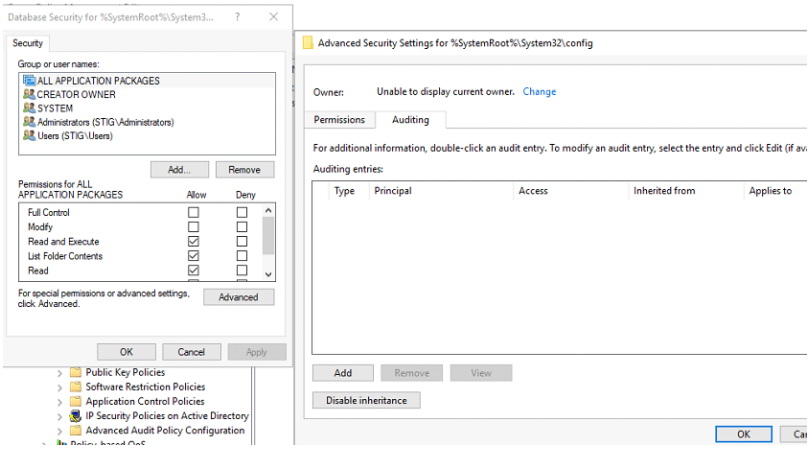
1. Right click File System, click Add File.



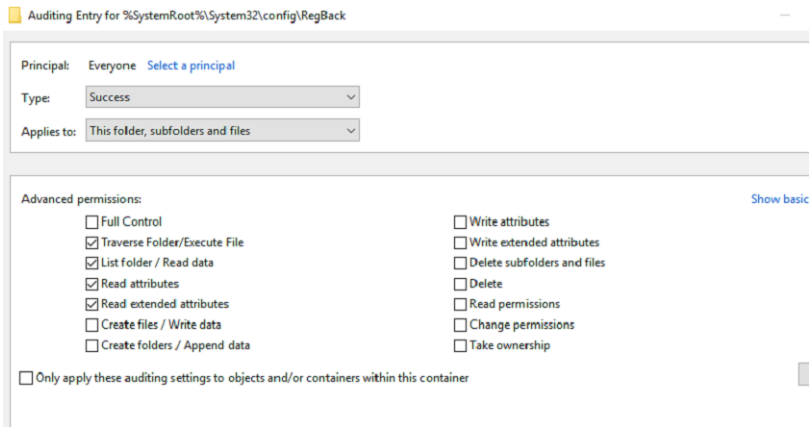
2. Add the folder path *%SystemRoot%\System32\Config*



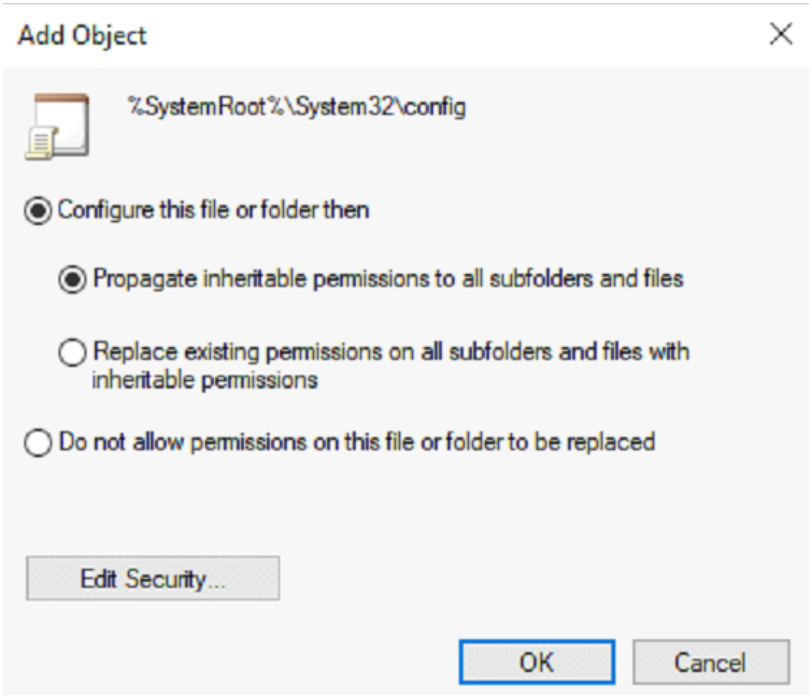
3. Click Advanced, then click Auditing tab, and click Add.



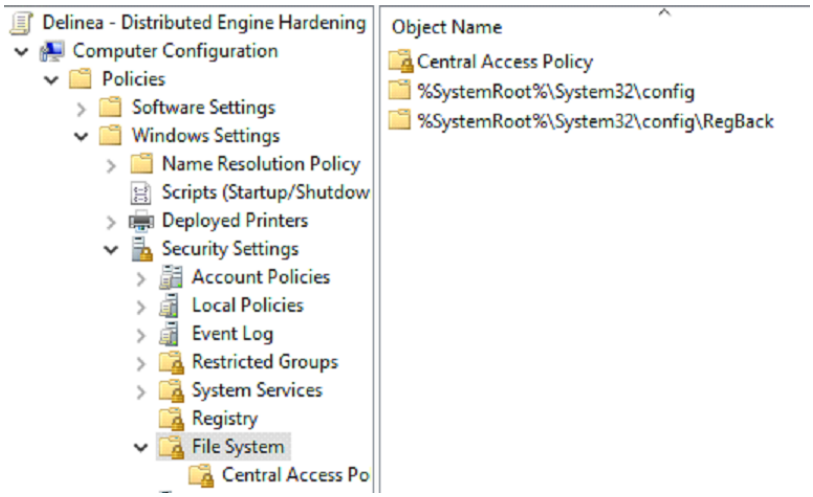
- Change Principal to Everyone, select Show Advanced Permissions, and select the following boxes:
- Traverse Folder\ Execute File
- List folder\ Read data
- Read attributes
- Read extended attribute



4. Click OK, then click Apply Settings.



5. Perform the same steps above for *SystemRoot%\System32\Config\RegBack*



Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies/Security Options

Policy	Recommended Value	Comment or Vulnerability
Accounts: Administrator account status	Enabled	
Accounts: Guest account status	Disabled	
Accounts: Limit local account use of blank passwords to console logon only	Enabled	
Accounts: Rename administrator account	It is recommended to change both the Administrator and the guest names to a name that will not testify about their permissions, and also to create a new locked and unprivileged user name Administrator as bate	Comment: Apply this parameter according to the organization security policy. Vulnerability: The administrators default name is known as a high privilege user. This user is a target for hacking attempts. Severity of the damage: Medium Operational aspects: None
Audit: Audit the use of Backup and Restore privilege.	Enabled	Vulnerability: The system does not monitor backup and restore activities of files, therefore it does not allow exposing unusual activities in this area. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Devices: Allowed to format and eject removable media	Administrator	Vulnerability: Users may be able to move data on removable disks to a different computer where they have administrative privileges. The user could then take ownership of any file, grant themselves full control, and view or modify any file. The fact that most removable storage devices will eject media by pressing a mechanical button diminishes the advantage of this policy setting. Severity of the damage: Low Operational aspects: None
Devices: Prevent users from installing printer drivers	Enabled	Vulnerability: A malicious user could install inappropriate printer drivers in a deliberate attempt to damage the computer, or a user might accidentally install malicious software that masquerades as a printer driver. Severity of the damage: Low Operational aspects: None
Domain member: Disable machine account password changes	Disabled	Vulnerability: Computers that cannot automatically change their account password are at risk from an attacker who could determine the password for the computer's domain account. Severity of the damage: Low Operational aspects: None
Domain member: Maximum machine account password age	30 days	Vulnerability: Setting this parameter to 0 will allow an attacker to execute Brute Force attacks to find the computer account password. Severity of the damage: Low Operational aspects: None
Domain member: Require strong (Windows 2000 or later) session key	Enabled	Vulnerability: Session keys that are used to establish secure channel communications between domain controllers and member computers are much stronger in Windows 2000 than they were in previous Windows operating systems. Severity of the damage: Low Operational aspects: None
Interactive logon: Do not display last user name	Enabled	Vulnerability: An attacker with access to the console (for example, someone with physical access or someone who is able to connect to the server through Remote Desktop Services, also known as Terminal Services) could view the name of the last user who logged on to the server. The attacker could then try to guess the password, use a dictionary, or use a brute force attack to try to log on. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Interactive logon: Do not require CTRL+ALT+DEL	Disabled	Vulnerability: If users are not required to press CTRL+ALT+DEL, they are susceptible to attacks that attempt to intercept their passwords, If this setting is enabled, an attacker could install a Trojan horse program that looks like the standard logon dialog box in the Windows operating system, and capture the user's password. Severity of the damage: Low Operational aspects: None
Interactive logon: Number of previous logons to cache (in case domain controller is not available).	0	Vulnerability: Users who access the server console will have their logon credentials cached on that server. An attacker who is able to access the file system of the server could locate this cached information and use a brute force attack to attempt to determine user passwords. Severity of the damage: Medium Operational aspects: The local Administrator password should be known in case of DC unavailability.

Policy	Recommended Value	Comment or Vulnerability
Interactive logon: Require Domain Controller authentication to unlock workstation	Enabled	Vulnerability: By default, the computer caches in memory the credentials of any users who are authenticated locally. The computer uses these cached credentials to authenticate anyone who attempts to unlock the console. When cached credentials are used, any changes that have recently been made to the account-such as user rights assignments, account lockout, or the account being disabled-are not considered or applied after the account is authenticated. User privileges are not updated, and (more important) disabled accounts are still able to unlock the console of the computer. Severity of the damage: Medium Operational aspects: The local Administrator password should be known in case of DC unavailability
Microsoft network client: Send unencrypted password to third-party SMB servers.	Disabled	Vulnerability: The server can transmit passwords in plaintext across the network to other computers that offer SMB services. These other computers might not use any of the SMB security mechanisms that are included with Windows Server 2003 and above. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Microsoft network server: Amount of idle time required before suspending session	15 minutes	Vulnerability: Each SMB session consumes server resources, and numerous null sessions will slow the server or possibly cause it to fail. An attacker could repeatedly establish SMB sessions until the server's SMB services become slow or unresponsive. Severity of the damage: Medium Operational aspects: None
Microsoft network server: Attempt S4U2Self to obtain claim information	Disabled	Vulnerability: Enabling this policy setting allows you take advantage of features in Windows Server 2012 and Windows 8 for specific scenarios to use claims-enabled tokens to access files or folders that have claim-based access control policy applied on Windows operating systems prior to Windows Server 2012 and Windows 8. Severity of the damage: Medium Operational aspects: None
Microsoft network server: Server SPN target name validation level	Off	Vulnerability: This policy setting controls the level of validation that a server with shared folders or printers performs on the service principal name (SPN) that is provided by the client computer when the client computer establishes a session by using the SMB protocol. The level of validation can help prevent a class of attacks against SMB servers (referred to as SMB relay attacks). This setting will affect both SMB1 and SMB2. Severity of the damage: Low Operational aspects: None
Network access: Do not allow anonymous enumeration of SAM accounts	Enabled	Vulnerability: An unauthorized user could anonymously list account names and shared resources and use the information to attempt to guess passwords or perform social engineering attacks. Severity of the damage: Medium Operational aspects: None
Network access: Do not allow anonymous enumeration of SAM accounts and shares	Enabled	Vulnerability: An unauthorized user could anonymously list account names and shared resources and use the information to attempt to guess passwords or perform social-engineering attacks. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Network access: Do not allow storage of passwords and credentials for network authentication.	Enabled	Vulnerability: Passwords that are cached can be accessed by the user when logged on to the computer. Although this information may sound obvious, a problem can arise if the user unknowingly runs malicious software that reads the passwords and forwards them to another, unauthorized user. Severity of the damage: Medium Operational aspects: This parameter could affect windows schedule task services
Network access: Let Everyone permissions apply to anonymous users	Disabled	Vulnerability: The system will allow all users, including users who have not identified themselves in the Domain, perform operations of reading information related to user accounts and the names of the shares. Severity of the damage: Medium Operational aspects: None
Network access: Named Pipes that can be accessed anonymously	List has been deleted	The policy was enabled and the existing list was deleted. Vulnerability: Ability to remotely access data on the system by an unauthorized user. Severity of the damage: Low Operational aspects: None
Network access: Remotely accessible registry paths	List has been deleted	The policy was enabled and the existing list was deleted. Vulnerability: An attacker could use information in the registry to facilitate unauthorized activities. To reduce the risk of such an attack, suitable ACLs are assigned throughout the registry to help protect it from access by unauthorized users. Severity of the damage: Low Operational aspects: None
Network access: Remotely accessible registry paths and subpaths	List has been deleted	The policy was enabled and the existing list was deleted. Vulnerability: An attacker could use information in the registry to facilitate unauthorized activities. To reduce the risk of such an attack, suitable ACLs are assigned throughout the registry to help protect it from access by unauthorized users. Severity of the damage: Low Operational aspects: None
Network access: Restrict anonymous access to Named Pipes and Shares.	Enabled	Vulnerability: Null sessions are a weakness that can be exploited through shared folders on computers environment. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Network access: Shares that can be accessed anonymously	List has been deleted	The policy was enabled and the existing list was deleted. Vulnerability: Any shared folders that are listed can be accessed by any network user, which could lead to the exposure or corruption of sensitive data. Severity of the damage: Medium Operational aspects: None
Network access: Sharing and security model for local accounts	Classic - Local users authenticate as themselves	Vulnerability: With the Guest only model, any user who can authenticate to the server over the network does so with guest privileges, which means that they will not have write access to shared resources on that server. Although this restriction does increase security, it makes it more difficult for authorized users to access shared resources on the server because ACLs on those resources must include access control entries (ACEs) for the Guest account. With the Classic model, local accounts should be password protected. Otherwise, if Guest access is enabled, anyone can use those user accounts to access shared system resources. Severity of the damage: Low Operational aspects: None
Network security: Do not store LAN Manager hash value on next password change	Enabled	Vulnerability: The SAM file can be targeted by attackers who seek access to user name and password hashes. Such attacks use special tools to discover passwords, which can then be used to impersonate users and gain access to resources on your network. Severity of the damage: Medium Operational aspects: None
Network security: Force logoff when logon hours expire	Enabled	Vulnerability: Users can remain connected to the computer outside of their allotted logon hours. Severity of the damage: Low Operational aspects: None
Network security: LAN Manager authentication level	Send NTLMv2 Responses Only/Refuse LM & NTLM	Vulnerability: The system allows identification of users in the old LM and NTLM protocols. The old identification protocols are vulnerable to attacks. Severity of the damage: Medium Operational aspects: These parameters could effect on legacy system if the system don't support NTLMv2

Policy	Recommended Value	Comment or Vulnerability
Network security: Minimum session security for NTLM SSP based (including secure RPC) clients	Require NTLMv2 session security Require 128-bit encryption	Vulnerability: Network traffic that uses the NTLM Security Support Provider (NTLM SSP) might be exposed such that an attacker who has gained access to the network can create man-in-the-middle attacks. Severity of the damage: Medium Operational aspects: None
Network security: Minimum session security for NTLM SSP based (including secure RPC) servers	Require NTLMv2 session security Require 128-bit encryption	Vulnerability: Network traffic that uses the NTLM Security Support Provider (NTLM SSP) might be exposed such that an attacker who has gained access to the network can create man-in-the-middle attacks. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Recovery console: Allow automatic administrative logon	Disabled	Vulnerability: The Recovery Console can be very useful when you need to troubleshoot and repair computers that do not start. However, it is dangerous to allow automatic logon to the console. Anyone could walk up to the server, disconnect the power to shut it down, restart it, select Recover Console from the Restart menu, and then assume full control of the server. Severity of the damage: Medium Operational aspects: None
Recovery console: Allow floppy copy and access to all drives and all folders	Disabled	Vulnerability: An attacker who can cause the system to restart into the Recovery Console could steal sensitive data and leave no audit or access trail. Severity of the damage: Low Operational aspects: None
Shutdown: Allow system to be shut down without having to log on	Disabled	Vulnerability: Users who can access the console locally could shut down the computer. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Shutdown: Clear virtual memory pagefile	Enabled	Vulnerability: Important information that is kept in real memory may be written periodically to the paging file to help the operating system handle multitasking functions. An attacker who has physical access to a server that has been shut down could view the contents of the paging file. The attacker could move the system volume into a different computer and then analyze the contents of the paging file. Although this process is time consuming, it could expose data that is cached from random access memory (RAM) to the paging file. Severity of the damage: Low Operational aspects: It takes longer to shut down and restart the computer, especially on computers with large paging files.
System Settings: Optional subsystems	No one	Enable the policy and delete the existing list of users that will be populated by default. Vulnerability: The POSIX subsystem introduces a security risk that relates to processes that can potentially persist across logons. If a user starts a process and then logs out, there is a potential that the next user who logs on to the computer could access the previous user's process. This would allow the second user to take actions on the process by using the privileges of the first user. Severity of the damage: Low Operational aspects: None
System settings: Use Certificate Rules on Windows Executables for Software Restriction Policies	Enable	Vulnerability: Without the use of software restriction policies, users and computers might be exposed to the running of unauthorized software, such as viruses and Trojans horses. Severity of the damage: Medium Operational aspects: None
User Account Control: Use Admin Approval Mode for the built-in Administrator account	Enable	Vulnerability: Malicious software running under elevated credentials without the user or administrator being aware of its activity. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
User Account Control: Allow UIAccess applications to prompt for elevation without using the secure desktop	Disable	Vulnerability: Without the use of software restriction policies, users and computers might be exposed to the running of unauthorized software, such as viruses and Trojans horses. Severity of the damage: Medium Operational aspects: None
User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode	Prompt for consent for non-Windows binaries	Vulnerability: Malicious programs running under elevated credentials without the user or administrator being aware of their activity. This setting raises awareness to the user that a program requires the use of elevated privilege operations and requires that the user be able to supply administrative credentials in order for the program to run. Severity of the damage: Medium Operational aspects: None
User Account Control: Behavior of the elevation prompt for standard users	Prompt for credentials on the secure desktop	Vulnerability: Malicious programs running under elevated credentials without the user or administrator being aware of their activity. This setting raises awareness to the user that a program requires the use of elevated privilege operations and requires that the user be able to supply administrative credentials in order for the program to run. Severity of the damage: Low Operational aspects: None
User Account Control: Run all administrator in admin approval mode	Enable	Vulnerability: This is the setting that turns on or off UAC. If this setting is disabled, UAC will not be used and any security benefits and risk mitigations that are dependent on UAC will not be present on the system. Severity of the damage: Medium Operational aspects: None
User Account Control: Switch to the secure desktop when prompting for elevation	Enable	Vulnerability: Elevation prompt dialog boxes can be spoofed, causing users to disclose their passwords to malicious software. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
User Account Control: Virtualize file and registry write failures to per-user locations	Enable	Severity of the damage: Low Operational aspects: None

Computer Configuration > Administrative Templates > Windows Components > Security Settings > Remote Desktop Services

Vulnerability:

An unlimited number of open connections can cause denial of Service attack on the Remote Desktop services, also known as Terminal Services.

If a disconnected session kept alive that can lead a session hijacking by an attacker.

Clipboard mapping enables the client to transfer a virus or a malicious application to the server as well as copy configuration or sensitive data from the server back to the client machine. There is a risk of infecting to the whole network or damaging the system.

Severity of the damage:

Medium

Operational aspects:

None

Path	Policy	Recommended Value
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Connections	Automatic reconnection	Disabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Connections	Configure keep-alive connection interval	Enabled Keep-Alive interval:1
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Connections	Deny logoff of an administrator logged in to the console session	Enabled

Path	Policy	Recommended Value
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection	Do not allow Clipboard redirection	Disabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection	Do not allow supported Plug and Play device redirection	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection	Do not allow COM port redirection	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection	Do not allow LPT port redirection	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection	Do not allow drive redirection	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Security	Do not allow local administrators to customize permissions	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Temporary Folders	Do not delete temp folders upon exit	Disabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Temporary Folders	Do not use temporary folders per session	Disabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Session Time Limits	End session when time limits are reached	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Remote Session Environment	Remove "Disconnect" option from Shut Down dialog	Enabled

Path	Policy	Recommended Value
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Remote Session Environment	Remove Windows Security item from Start menu	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Security	Require secure RPC communication	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Security	Set client connection encryption level	Enabled Encryption Level: High Level
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Connections	Set rules for remote control of Remote Desktop Services user sessions	Enabled View Session without user's permission
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Session Time Limits	Set time limit for active but idle Remote Desktop Services sessions	Enabled
Computer Configuration > Policies > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Session Time Limits	Set time limit for disconnected sessions	Enabled 15 minutes

Computer Configuration > Policies > Windows Settings > Security Settings > User Rights Assignment

Policy	Recommended Value	Comment
Access Credential Manager as a trusted caller		Vulnerability: If an account is given this right, the user of the account can create an application that calls into Credential Manager and is then provided the credentials for another user. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment
Access this computer from the network	BUILTIN\Administrators	Vulnerability: This right allows the users to use the SMB communications protocol in front of the server. This protocol allows access to the operating resources, such as: sharing and remote system administration using the operating system's built-in tools. Severity of the damage: Medium Operational aspects: None
Act as part of the operating system		Vulnerability: Users with the Act as part of the operating system user right can take complete control of the computer and erase evidence of their activities. Severity of the damage: Medium Operational aspects: None
Adjust memory quotas for a process	NT AUTHORITY\NETWORK SERVICE, NT AUTHORITY\LOCAL SERVICE, BUILTIN\Administrators	Vulnerability: A user with the Adjust memory quotas for a process user right can reduce the amount of memory that is available to any process, which could cause business-critical network applications to become slow or to fail. This privilege could be used to start a denial-of-service (DoS) attack. Severity of the damage: Medium Operational aspects: None
Allow log on locally	BUILTIN\Administrators	Vulnerability: Any account with the Allow log on locally user right can log on at the console of the computer. If you do not restrict this user right to legitimate users who must log on to the console of the computer, unauthorized users could download and run malicious software to elevate their privileges. Severity of the damage: Medium Operational aspects: None
Allow log on through Remote Desktop Services	BUILTIN\Administrators	Vulnerability: Any account with the Allow log on through Remote Desktop Services user right can log on to the remote console of the computer. If you do not restrict this user right to legitimate users who must log on to the console of the computer, unauthorized users could download and run malicious software to elevate their privileges. Severity of the damage: Medium Operational aspects: None
Back up files and directories	BUILTIN\Administrators	Vulnerability: Users who can back up data from a computer could take the backup media to a non-domain computer on which they have administrative privileges and restore the data. They could take ownership of the files and view any unencrypted data that is contained within the backup set. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment
Bypass traverse checking	NT AUTHORITY\ NETWORK SERVICE, NT AUTHORITY\LOCAL SERVICE, NT AUTHORITY\Authenticated Users, BUILTIN\Administrators	Vulnerability: This right allows the user to access files and partitions although he is not authorized to view files and change them. Severity of the damage: Medium Operational aspects: None
Change the system time	BUILTIN\Administrators, NT AUTHORITY\LOCAL SERVICE	Vulnerability: Users who can change the time on a computer could cause several problems. For example, time stamps on event log entries could be made inaccurate, an attacker who changes a computer's time may cause that computer to be unable to obtain or grant Kerberos protocol tickets. Severity of the damage: Medium Operational aspects: None
Change the time zone	BUILTIN\Administrator	Vulnerability: Changing the time zone represents little vulnerability because the system time is not affected. This setting merely enables users to display their preferred time zone while being synchronized with domain controllers in different time zones. Severity of the damage: Low Operational aspects: None
Create a token object		Vulnerability: A user account that is given this user right has complete control over the system, and it can lead to the system being compromised. Severity of the damage: High operational aspects: None
Create global objects	NT AUTHORITY\SERVICE, NT AUTHORITY\NETWORK SERVICE, NT AUTHORITY\LOCAL SERVICE, BUILTIN\Administrators	Vulnerability: Users who can create global objects could affect processes that run under other users' sessions. This capability could lead to a variety of problems, such as application failure or data corruption. Severity of the damage: Medium Operational aspects: None
Create permanent shared objects		Vulnerability: Users who have the Create permanent shared objects user right could create new shared objects and expose sensitive data to the network. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment
Create symbolic links	Administrators	Vulnerability: Users who have the Create symbolic links user right could inadvertently or maliciously expose your system to symbolic link attacks. Symbolic link attacks can be used to change the permissions on a file, to corrupt data, to destroy data, or as a DoS attack. Severity of the damage: Low Operational aspects: None
Debug programs	BUILTIN\Administrator	Vulnerability: The Debug programs user right can be exploited to capture sensitive computer information from system memory or to access and modify kernel or application structures. Some attack tools exploit this user right to extract hashed passwords and other private security information or to insert rootkit code. Severity of the damage: Low Operational aspects: None
Deny access to this computer from the network	BUILTIN\Guests	Vulnerability: Users who can log on to the computer over the network can enumerate lists of account names, group names, and shared resources. Users with permission to access shared folders and files can connect over the network and possibly view or modify data. Severity of the damage: Medium Operational aspects: None
Deny log on as a batch job	BUILTIN\Guests	Vulnerability: Accounts that have the Deny log on as a batch job user right could be used to schedule jobs that could consume excessive computer resources and cause a DoS condition. Severity of the damage: Medium Operational aspects: None
Deny log on as a service	BUILTIN\Guests	Vulnerability: Accounts that can log on as a service could be used to configure and start new unauthorized services, such as a keylogger or other malicious software. Severity of the damage: Medium Operational aspects: None
Deny log on locally	BUILTIN\Guests	Vulnerability: Any account with the ability to log on locally could be used to log on at the console of the computer. If this user right is not restricted to legitimate users who must log on to the console of the computer, unauthorized users might download and run malicious software that elevates their privileges. Severity of the damage: Medium Operational aspects: None

Policy	Recommended Value	Comment
Deny log on through Remote Desktop Services	BUILTIN\Guests	Vulnerability: Any account with the right to log on through Remote Desktop Services could be used to log on to the remote console of the computer. If this user right is not restricted to legitimate users who need to log on to the console of the computer, malicious users might download and run software that elevates their privileges. Severity of the damage: Medium Operational aspects: None
Enable computer and user accounts to be trusted for delegation	BUILTIN\Administrators	Vulnerability: Misuse of the Enable computer and user accounts to be trusted for delegation user right could allow unauthorized users to impersonate other users on the network. An attacker could exploit this privilege to gain access to network resources and make it difficult to determine what has happened after a security incident. Severity of the damage: Medium Operational aspects: None
Force shutdown from a remote system	BUILTIN\Administrators	Vulnerability: Any user who can shut down a computer could cause a DoS condition to occur. Therefore, this user right should be tightly restricted. Severity of the damage: Low Operational aspects: None
Generate security audits	NT AUTHORITY\NETWORK SERVICE, NT AUTHORITY\LOCAL SERVICE	Vulnerability: Accounts that can write to the Security log could be used by an attacker to fill that log with meaningless events. If the computer is configured to overwrite events as needed, attackers could use this method to remove evidence of their unauthorized activities. If the computer is configured to shut down when it is unable to write to the Security log and it is not configured to automatically back up the log files, this method could be used to create a DoS condition. Severity of the damage: Low Operational aspects: None
Increase scheduling priority	BUILTIN\Administrators	Vulnerability: Increasing the working set size for a process decreases the amount of physical memory that is available to the rest of the system. Severity of the damage: Low Operational aspects: None
Load and unload device drivers	BUILTIN\Administrators	Vulnerability: Device drivers run as highly privileged code. A user who has the Load and unload device drivers user right could unintentionally install malicious software that masquerades as a device driver. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Lock pages in memory	BUILTIN\Administrators	Vulnerability: Users with the Lock pages in memory user right could assign physical memory to several processes, which could leave little or no RAM for other processes and result in a DoS condition. Severity of the damage: Low Operational aspects: None
Manage auditing and security log	BUILTIN\Administrators	Vulnerability: Anyone with the Manage auditing and security log user right can clear the Security log to erase important evidence of unauthorized activity. Severity of the damage: Medium Operational aspects: None
Modify an object label		Vulnerability: Anyone with the Modify an object label user right can change the integrity level of a file or process so that it becomes elevated or decreased to a point where it can be deleted by lower-level processes. Either of these states effectively circumvents the protection offered by Windows Integrity Controls and makes your system vulnerable to attacks by malicious software. Severity of the damage: Low Operational aspects: None
Modify firmware environment values	BUILTIN\Administrators	Vulnerability: Anyone who is assigned the Modify firmware environment values user right could configure the settings of a hardware component to cause it to fail, which could lead to data corruption or a DoS condition. Severity of the damage: Medium Operational aspects: None
Perform volume maintenance tasks	BUILTIN\Administrators	Vulnerability: A user who is assigned the Perform volume maintenance tasks user right could delete a volume, which could result in the loss of data or a DoS condition. Also, disk maintenance tasks can be used to modify data on the disk such as user rights assignments that might lead to escalation of privileges. Severity of the damage: Low Operational aspects: None
Profile single process	BUILTIN\Administrators	Vulnerability: The Profile single process user right presents a moderate vulnerability. Attackers with this user right could monitor a computer's performance to help identify critical processes that they might want to attack directly. Attackers may be able to determine what processes run on the computer so that they could identify countermeasures that they may need to avoid, such as antivirus software or an intrusion-detection system. They could also identify other users who are logged on to a computer. Severity of the damage: Low Operational aspects: None

Policy	Recommended Value	Comment or Vulnerability
Restore files and directories	BUILTIN\Administrators	Vulnerability: An attacker with the Restore files and directories user right could restore sensitive data to a computer and overwrite data that is more recent, which could lead to loss of important data, data corruption, or a denial-of-service condition. Attackers could overwrite executable files that are used by legitimate administrators or system services with versions that include malicious software to grant themselves elevated privileges, compromise data, or install programs that provide continued access to the computer. Severity of the damage: Medium Operational aspects: None
Shut down the system	BUILTIN\Administrators	Vulnerability: The ability to shut down the server should be limited to a very small number of trusted administrators. Severity of the damage: Low Operational aspects: None
Take ownership of files or other objects	BUILTIN\Administrators	Vulnerability: Any users with the Take ownership of files or other objects user right can take control of any object, regardless of the permissions on that object, and then make any changes that they want to make that object. Such changes could result in exposure of data, corruption of data, or a DoS condition. Severity of the damage: High Operational aspects: None

Security Hardening Guide

This topic only applies to **Secret Server On-Premises**.

Introduction

This document outlines security hardening for securing your Secret Server instance, whether it be installed on a single server or in a multi-clustered environment.

Throughout this guide, many references are made to "configuration" settings. Unless otherwise specified, this refers to the settings found by selecting **Configuration** from the **Admin** menu in Secret Server.

Overview

It is critical to secure your Secret Server implementation. That needs to include a layered approach to security (defense in depth), including the operating system, software updates, physical access, protocols, system settings, backups, and personnel procedures. This section of the guide links to other sections containing more details.

Best Practices

General

- **Keep Windows up-to-date:** Microsoft regularly releases security patches that resolve vulnerabilities in Windows operating systems.
- **Backup at least daily:** Consider your disaster recovery plan. See "Backup and Disaster Recovery" on page 451.
- **Review system log for errors:** Periodically check the system log (Admin > System Log) for recurring errors. Also do so after any upgrades.
- **Whole-disk encryption:** Use whole disk encryption, such as BitLocker, with a trusted platform module (TPM) to prevent those with physical access from removing disks to gain access to your Secret Server application by circumventing OS and application authentication.
- **Security Hardening Standards:** Consider security hardening standards that apply to either the operating system or applications, such as IIS or Microsoft SQL. Secret Server is compatible with CIS Level 1 and CIS Level 2 hardening and has STIG compatibility.

> Attaining full security-hardening standards compatibility is a Delinea priority.

Active Directory

On Active Directory domain controllers, there is a set of unsafe default configurations for LDAP channel binding that allow LDAP clients to communicate with them without ensuring LDAP channel binding and LDAP signing. This can open the controllers to privilege vulnerabilities. See 2020 LDAP channel binding and LDAP signing requirements for Windows for details.

Database

- **Limit access to your Secret Server database:** When you create your Secret Server database, limit access to as few users as possible. We recommend you disable the "sa" account in the SQL instance that contains Secret Server.
- **Limit access to other databases:** When you create a database account for SS, you should ensure it only has access to the Secret Server database.
- **Use Windows Authentication for database access:** Windows authentication is much more secure than SQL authentication. See Choose an Authentication Mode (TechNet article) for details. To use Windows authentication in SS, you need to create a service account. See the "Accessing MS SQL Server with IWA" on page 1419 for details.
- **Limit access to your database backups:** Database backups are critical for disaster recovery, but they also carry a risk if someone gains access. The Secret Server database is encrypted, but you should still limit access to ensure maximum security. Limit access to database backups to as few users as possible.
- **Don't share a SQL instance with less secure databases:** Putting the database on a server with less-secure database instances can expose vulnerabilities. For example, an attacker could use SQL injection on another application to access your private Secret Server database. If you intend to put Secret Server on a shared SQL

instance, ensure that the other databases are classified internally as sensitive as Secret Server and have similar security controls in place.

- **Review Microsoft's recommendations for SQL security:** See the Securing SQL Server article in Microsoft's documentation.

> Secret Server also supports SQL Server Transparent Data Encryption (TDE) for further protection of the database files. This can have a slight performance impact on the environment and can increase the complexity of the database configuration. Please review this page for more information: Transparent Data Encryption (TDE).

Application Server

- **Use SSL (HTTPS):** We require using Secure Sockets Layer (SSL) encryption to ensure that all communication between the Web browser and Secret Server is secure. We recommend you install a third-party certificate, domain certificate, or self-signed certificate on your website. For information on creating and installing a self-signed certificate, please see "Installing Self-Signed SSL Certificates" on page 430.
- **Force SSL (HTTPS):** Even after you install an SSL certificate, users may still be able to access Secret Server through regular HTTP. To that, enable the "Force HTTPS/SSL" option in Secret Server at Admin > Configuration on the **Security** tab.
- **Limit access to your Secret Server directory**. This contains the Secret Server encryption key, as well as the database connection information (these values are encrypted but remember "defense in depth." Try to grant access to as few users as possible).
- **Limit logon rights to the application server**. Administrators accessing the Application Server directly could attempt to monitor memory in use on the server. Secret Server does several things to protect application memory but the best safeguard is to limit access to the Application Server to as few users as possible.
- **Protect your encryption key**. The encryption key for Secret Server is contained in the encryption.config file, which resides in your Secret Server directory. This file is obfuscated and encrypted, but "defense in depth" would require limiting access to the file. Using DPAPI to encrypt your encryption.config file is one option. This will use machine-specific encryption to encrypt the file. Make sure you back up the original file before enabling this option. To further protect the file, you can enable EFS encryption. EFS (Encrypting File System) is a Microsoft technology that allows a user or service account to encrypt files with login passwords. For more details, read Protecting Your Encryption Key Using EFS in this same article. The most secure option is to use a Hardware Security Module (HSM) to protect the Secret Server encryption key. For more information, see "Using Hardware Security Modules " on page 1440.

Application Settings

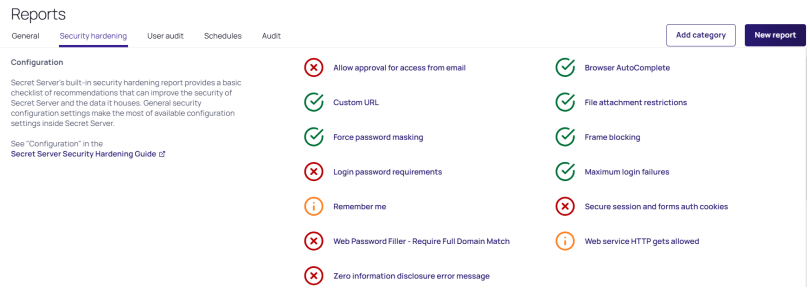
- **Use doublelock for your most sensitive secrets:** DoubleLock is a feature in Secret Server that allows secrets to be protected with additional AES256 encryption keys. Each user gets their own public and private key set when using doublelock. Their private key is protected by an additional password (user-specific, not a shared password) that each user must enter when using doublelock. DoubleLock protects from situations where you accidentally assign someone to the wrong AD group or an attacker gains full access to both your database and Web server - they still will not be able to access doublelocked secrets. For more information, refer to "QuantumLock Overview" on page 1080.

- **Secure the local admin account:** When you create the first user in SS, it is a privileged admin account that you can use when your domain is down. We recommend that you choose a non-obvious name for this account and protect it with a very strong password. This password should be stored in a physical safe with limited access (there is no need to use this account except in emergencies where other accounts are not working if AD is down or some other reason).
- **Review activity reports:** It is a good practice to regularly review the activity and permissions reports. This can help find anomalies in secret permissions and login failures.
- **Use event subscriptions or SIEM to notify of any security anomalies:** Use event subscriptions to send email alerts on various events in the system, and syslog can send events to a SIEM tool for correlation. For example, this could be used to notify administrators if there are failed login attempts or if certain secrets are viewed.

Security Hardening Report

Secret Server contains a built-in security hardening report to provide a basic checklist of recommendations that can improve the security of Secret Server and the data it houses. The items in this report range from common tasks, such as ensuring SSL is configured, to more advanced options like DPAPI encryption of the encryption key. To find this report, click the **Reports** on the dartboard, and then select the **Security Hardening** tab.

Figure: Security Harding Report:



An X denotes a failure, and a checkmark denotes a pass. An exclamation point is a warning. Typically, Secret Server could not detect a setting or all aspects of a check were not completed. For example, TLS 1.0 was disabled but TLS 1.1 was not.

You will find the following items in the report:

> The individual items below are in alphabetical order, not the order they are in the Hardening Report. The sections are in the same order as the report. This was because the report name does not always match the name of the corresponding label on the configuration UI control. In addition, the controls are not in the same order in the UI as their equivalents in the report.

Configuration Section

Allow Approval for Access from Email

Recommendation: Off

Allow Approval For Access from Email is a convenience option that allows users to approve or deny a secret access request by clicking a link in the request email sent by Secret Server. Allow Approval From Email does not require a

user to authenticate with Secret Server when approving access to a secret. This can be a security concern if the approver's email account becomes compromised, which could allow an attacker to mitigate MFA in some cases to complete an approval. Turn Allow Approval From Email off to get a pass result.

To disable this setting, find the **Permission Options** section of the **Configuration Settings** page and disable **Allow Approval for Access from Email**.

Custom URL

Recommendation: On

The Custom URL sets a definitive URL for Secret Server. Without it, certain features in Secret Server which need to build a link back to the server must construct the link using the host value on the request, which is susceptible to manipulation.

To disable this setting, go to **Admin > Security hardening**. Under the **Security hardening** tab, select **Custom URL**.

File Attachment Restrictions

Recommendation: On


Labeled **Enable File Restrictions** in the UI.

File attachment restrictions allows administrators to configure what kind of file attachments can be uploaded to secrets. This helps protect users from being tricked into downloading a malicious secret attachment. The file extension and maximum file size can be specified, such as:

```
*.7z, *.bmp, *.ca-bundle, *.cer, *.config, *.crt, *.csr, *.csv, *.dat, *.doc, *.docx,
*.gif, *.gz, *.id-rsa, *.jpeg, *.jpg, *.json, *.key, *.lic, *.p7b, *.pcf, *.pdf, *.pem,
*.pfx, *.pkey, *.png, *.ppk, *.pub, *.tar, *.tif, *.tiff, *.tpm, *.txt, *.vdx, *.vsd,
*.vsdx, *.xls, *.xlsx, *.xml, *.zip
```

This security check will fail if the file attachment restrictions is not enabled. This check will return warnings if a potentially dangerous file extension is allowed, maximum file size is not specified, or maximum file size is greater than 30 MB.

Go to **Admin > Configuration > Security tab > File Restrictions section** to change these settings.

Frame Blocking

Recommendation: On


Labeled **Enable Frame Blocking** in the UI.

Do not allow Secret Server to be opened in a `<iframe>` HTML tag on another, potentially malicious, site. This adds the HTTP header `X-Frame-Options: DENY`. This deters clickjacking and spreading potential XSS vulnerabilities.

Go to **Admin > Configuration > Frame Blocking** to change this setting.

Force Password Masking

Recommendation: On

Setting: Same

Password masking prevents over the shoulder viewing of your passwords by a casual observer (passwords show as *******). Note the number of asterisks does not relate to the length of the password for added security.

As an administrator, you can force all the secret password fields in the system to be masked when viewed. To do this, enable **Force Password Masking** on the **Configuration Settings** page. Only secret fields marked as a password type field on the secret template will be masked. There is also a user preference setting which will force password masking on all secret password fields viewed by the user.

This **Mask passwords when viewing Secrets** setting is found in the **Tools** > **Preferences** section for each user.

> If the "Force Password Masking" configuration setting discussed above is enabled, this user preference setting will be overridden and cannot be disabled.

Login Password Requirements

Passwords used by local users to log onto Secret Server can be strengthened by requiring a minimum length and using a variety of character sets. We recommend a minimum password length of eight characters. In addition, all character sets (lowercase, uppercase, numbers, and symbols) are required to get a pass result.

Turn on these login password settings on the **Local User Passwords** tab of the **Configuration Settings** page.

Maximum Login Failures

Recommendation: Reference the lockout policy for your organization. Most often, this setting will mirror the AD GPO lockout policy.

The maximum number of login failures is the number of attempts that can be made to log into Secret Server as a user before that user's account is locked. A user with the administer users role permission will then be required to unlock the user's account. The maximum failures allowed should be set to five or less to get a pass result.

Change the **Maximum Login Failures** setting on the **Login** tab of the **Configuration** settings.

Remember Me

Recommendation: Off

> Labeled **Allow Remember Me** in the UI.

"Remember Me" is a convenience option that allows users to remain logged onto Secret Server for up for a specific period. This setting can be a security concern because it does not require re-entry of credentials to gain access to Secret Server.

Disable **Allow Remember Me** on the **Login** tab of the **Configuration** page to get a pass result. It must be set to be valid for 1 day or less to not get a fail result.

> Closing a browser completely (all tabs) will log the user out of Secret Server, regardless of this setting.

Secure Session and Forms Auth Cookies

Secure Session and Forms *Authentication* Cookies.

Recommendation: See KBA

Cookies contain potentially sensitive information that can allow users to log onto application. By default, cookies are not marked with the secure attribute. That is, **they are transmitted unencrypted when a user accesses Secret Server through HTTP instead of HTTPS**.

For more information about how to secure your cookies, see "Securing ASP Cookies" on page 1457.

Web Service HTTP Gets Allowed (SOAP and REST-style)

Labeled **Allow HTTP Get** in the UI.

Recommendation: Off

Web service HTTP GET requests are allowed, which can be used for both SOAP and REST-style calls to many Secret Server Web service methods. This can be a security concern as a malicious user could create a link to the Web service that, when clicked, would execute the request. It is recommended to disable this feature to enhance security.

To mitigate security risks, disable the **Allow HTTP Get** option - navigate to **Admin > Webservices**, click **Edit**, uncheck **Allow HTTP GET** and click **Save**. This will prevent both SOAP and REST-style HTTP GET requests from being executed.

Zero Information Disclosure Error Message

Recommendation: On

Replace all error messages with a custom "contact your admin" message. Error messages can be very helpful when diagnosing installation and configuration issues. However, having errors displayed to a potential attacker can provide him or her with the critical information they need to perform a successful attack.

To hide error messages from the business user, add the `ZeroInformationDisclosureMessage` application setting to the `web-appSettings.config` file. This file is located in directory containing the Secret Server application files. Add the key below to this file in between the `<appSettings>` tags. The contents of that tag is displayed as a message that appears to the user whenever an error occurs in the system. For example:

```
<add key="ZeroInformationDisclosureMessage" value="An error occurred in the application.
Please contact your administrator." />
```

This setting is enabled by default in Secret Server 10.7.26+.

Database Section

SQL Account Using Least Permissions

Use the fewest Secret Server permissions as possible in the SQL Account used to access the database. We recommend using a least permission approach where the account only has dbOwner. See "SQL Server 2016

Standard Edition Installation" on page 111.

SQL Server Authentication Password Strength and Username

> This section addresses two separate but closely related settings: "SQL Server Authentication Password Strength" and "SQL Server Authentication Username."

Recommendation: Change settings as needed

SQL Server authentication requires a username and a strong password. Strong passwords are eight characters or longer and contain lowercase and uppercase letters, numbers, and symbols. In addition, the SQL Server authentication username should not be obvious.

You can change the credentials of a local SQL account through SQL Server Management Studio, where the Secret Server database is located. The SQL Server authentication credentials used by the application can then be changed by going to the installer `installer.aspx` page and changing them on step three. Using Windows authentication to authenticate to SQL Server is allowed.

For details about creating or modifying a SQL account for Secret Server, see "SQL Server and Secret Server" on page 97

Windows Authentication to Database

Recommendation: Configure

> If the SQL instance is *solely* using Windows authentication, this check will pass. If using mixed mode, it will fail—even you are using both Windows authentication plus SQL authentication.

Windows authentication takes advantage of Windows security to provide secure authentication to SQL Server. The SQL Server authentication options can be changed by going to the installer (installer.aspx) and changing them.

Environment Section

Application Pool Identity

Recommendation: Check configuration

The Application Pool identity appears to be a member of the administrators group on the system. This puts the system at risk by giving more access than necessary.

Check the identity of the application pool used by Secret Server in IIS. The Application Pool should be configured to use a service account and not be given unrestricted access to the server or domain.

DPAPI or HSM Encryption of Encryption Key

Recommendation: On

Encrypt your Secret Server encryption key, and limit decryption to that same server. Data Protection API (DPAPI) is an encryption library that is built into Windows operating systems. It allows encryption of data and configuration files based on the machine key. Enabling DPAPI Encryption in Secret Server protects the Secret Server encryption key by using DPAPI, so even getting access to the Secret Server encryption key is not enough to be useful—the

machine key is required. If you enable this option, back up your encryption key first, as a DPAPI encrypted file can only be used by the machine it was encrypted on.

To enable DPAPI encryption, to **Admin > Configuration > Security tab** and click the **Encrypt Key Using DPAPI** button.


This check also passes if Hardware Security Module (HSM) integration is enabled.

SSL Section

Require SMTP SSL

Recommendation: On


Labeled **Use SSL** (on the Email tab) in the UI.


We strongly recommend enabling this setting.

SMTP SSL is required to ensure that all communication between Secret Server and the email server is encrypted. Enable the "Use SSL" option in Secret Server to get a pass result.

Go to **Admin > Configuration > Email tab > Use SSL** to enable the setting.

Require SSL


Labeled **Force HTTPS/SSL** in the UI.

Recommendation: On


We **strongly** recommend using SSL for Secret Server.

Only use SSL (HTTPS) for Secret Server access. Secure Sockets Layer (SSL) is required to ensure that all communication between the Web browser and Secret Server is encrypted. To do so, you need an SSL certificate. You may use an existing wildcard certificate, create your own domain certificate, or purchase a third-party SSL certificate for the Secret Server website. For testing, you can use a self-signed certificate. See "Securing ASP Cookies" on page 1457 for more information.

Once the SSL certificate is installed, enable **Force HTTPS/SSL** on the **Security** tab of the **Configuration** page to force users to only access Secret Server over HTTPS and to receive a pass in the report.

SSL/TLS Hash

Recommendation: Confirm or remediate

Check the digest algorithm of the certificate. If the algorithm is SHA1, this check returns a warning because SHA1 is being phased out. If the digest algorithm is MD2, MD4, or MD5, the check will fail because they are not secure. SHA256, SHA384, and SHA512 will pass. This check fails if Secret Server cannot be loaded over HTTPS.

Example warning:

"The digest algorithm is sha1RSA, which is considered weak. The algorithm is being phased out and should be replaced with a better algorithm when it comes time to renew the SSL certificate."

Go to the browser's certificate information when logged onto Secret Server. This is usually a button next to the URL text box.

SSL/TLS Key

Recommendation: Confirm or remediate

Check the key size of the HTTPS certificate used. If it is RSA or DSA, the key must be at least 2048-bit to pass. If the signature algorithm of the certificate is ECDSA, the key size must be at least 256-bit to pass. If the algorithm of the certificate is unknown, the result shows "unknown. This check fails if Secret Server cannot be loaded over HTTPS.

Go to the browser's certificate information when logged onto Secret Server. This is usually a button next to the URL text box.

SSL/TLS Protocols

Recommendation: Confirm or remediate

Check for legacy SSL or TLS protocols, which should not be used in a secure environment. If the server accepts SSLv2 or SSLv3 connections, this check will fail. SSLv2 is not considered secure for data transport, and SSLv3 is vulnerable to the POODLE attack. If this server does not support TLSv1.1 or TLSv1.2, this check will give a warning because they are recommended. The SSL certificate used may affect what protocols can be used, even if they are enabled. This check will fail if Secret Server cannot be loaded over HTTPS.

You can check and modify these settings in the Window registry. See Transport Layer Security (TLS) Registry Settings in Microsoft's documentation.

Example warning:

"The server supports the accepts SSLv2 or SSLv3 connections protocol, which are weak. Consider disabling these protocols."

Using HTTP Strict Transport Security

Labeled **Enable HSTS** in the UI.

HTTP Strict Transport Security (HSTS) is an additional security layer for SSL. HSTS allows Secret Server, Password Reset Server, or Group Management Server to inform browsers that it should only be accessible over HTTPS. With this setting enabled, visitors are automatically redirected by their browser to the HTTPS-enabled site.

When the **Force HTTPS/SSL** option in enabled on the **Security** tab of the **Configuration** page, the **Enable HSTS** check box will be displayed. After the option is turned on, you can click **Advanced** to specify the maximum age in seconds for how long the policy should be in effect before re-evaluating. The default value is 31,540,000 seconds (a year)—the amount required to enable the checkmark, which assumes the site should never be accessed without TLS or SSL.

For details about this, see "Securing Traffic with HTTP Strict Transport Security" on page 785.

Security Settings Not in the Hardening Report

Apply TLS Certificate Chain Policy and Error Auditing

Recommendation: Confirm or remediate

Add audits for TLS certificate validation. Auditing will apply to all Active Directory domains using LDAPS and Syslog using TLS. Certificate policy options, including ignoring certificate revocation failures, apply to syslog using TLS only. The default is the most strict so the certificate chain policy may need to be updated. TLS errors will be logged to Security Audit Log found on the Administration page.

Disable the **Admin >Configuration > Security tab > Apply TLS Certificate Chain Policy and Error Auditing** setting.

TLS errors are logged to the **Security Audit** log at **Admin > See All > Security Audit Log**.

Enable FIPS Compliance

Recommendation: Off

Only allow FIPS-compliant encryption schemes. FIPS (Federal Information Processing Standards) is a set of standards for government entities. It covers many things, including encryption. For businesses, FIPS can be counter productive because it restricts them from using newer or improved existing encryption methods. In addition to enabling this setting, several other tasks are required to meet this standard, including enabling it for Windows itself. For more information, see "Enabling FIPS Compliance in Secret Server On-Premises" on page 1304 for details.

Go to **Admin > Configuration > Security tab > FIPS Compliance** to change this setting.

Key Rotation

Recommendation: Review KBA

> Key rotation is not a setting but an activity. It is included here for completeness (the entire Configuration Security tab)

Secret key rotation changes out the encryption key for secret data and re-encrypts that data with a new key. This helps you to meet compliance requirements mandating that encryption keys are changed on a regular basis. See "Secret Key Rotation" on page 1454.

Two-Factor Authentication

Users must authenticate to Secret Server at least once using either local Secret Server credentials or their Active Directory credentials. In addition, you can protect Secret Server by enabling two-factor authentication (2FA). 2FA is an additional security layer, such as a text message PIN code sent to your smart phone. The following options are supported by Secret Server for 2FA:

SAML

Secret Server supports the Security Assertions Markup Language (SAML), which provides a more centralized method of adding 2FA to the Secret Server log on. Please see the "Configuring SAML Single Sign-on" on page 422.

Email

Using email for 2FA means that after authenticating with their password, the user receives an email containing a one-time PIN code to enter. For this to work, an SMTP server must be configured in Secret Server and each user must have a valid email address associated with their account. For Active Directory users, the email address will be synced automatically from their domain account.

Check user email addresses at **Admin > Users**.

Soft Tokens

Soft tokens using the Time-based One-time Password (TOTP) algorithm, such as Google Authenticator and Microsoft Authenticator, are supported by Secret Server 2FA. Users are prompted to enter a token displayed on their mobile device each time they log onto Secret Server. The time-based token changes on a regular interval (such as 30 seconds).

RADIUS

One option is to use a Remote Authentication Dial-In User Service (RADIUS)compliant device, such as an RSA or CryptoCard token, as the second form of authentication. The user is prompted to enter his or her RADIUS password after initial authentication is done with their Secret Server or AD password.

To set this up, you first need to configure Secret Server to integrate with your RADIUS server, and then you can enable it for individual users or for by domain.

See "Enabling RADIUS Two-Factor Authentication" on page 440 for details.

Duo Security

Using this method requires that you have an active account for Duo Security. Duo Security provides several options for 2FA. The API hostname, integration key, and secret key values are required for Secret Server to authenticate Duo users.

See for details.

Enabling Two-Factor Authentication

Enabling for Users

To enable two-factor authentication for a user or several users at once, select **Users** from the **Admin** menu and then select the users in the grid. Use the bulk operation drop-down menu to choose the type of authentication to enable.

> If prerequisite settings are not configured, the 2FA option may be disabled or will not appear as an option. See the descriptions above for information about prerequisites for each type of two-factor authentication.

Enabling per Domain

Two-factor authentication can also be enabled per domain if you are syncing users from Active Directory. To do so, select **Active Directory** from the **Admin** menu and then click **Edit Domains**. Click the domain name and then click

Advanced (not required) to reveal the **Auto-Enable Two Factor for New Users** setting. Select this checkbox and click **Save and Validate**.

Roles

Secret Server uses role-based access control, which allows administrative and user capabilities to be partitioned by role. This can allow for granular control over which areas of the application a user has access to, for example, allowing someone the rights to manage licenses and view reports in Secret Server but nothing else.

Controlling Access to Features Using Roles

Limiting Role Access to the Export Permission

Exporting secrets from your Secret Server as text is very helpful for meeting regulations in certain industries (secrets can then be printed to paper and locked in a physical safe). It can also be used as another disaster recovery option, but access to exporting data from the Secret Server should be tightly controlled. You could create a separate role with just the export permission for anyone needing to export secrets.

Unlimited Administration Mode

Unlimited administration mode allows any role with the "unlimited administrator permission" to see all secrets in the Secret Server. This mode is very helpful for recovering passwords in emergencies or when staff are terminated. You can tightly control access to this feature by splitting out the role permissions for "administer configuration unlimited admin" and "unlimited administrator" into two different roles. This allows you to create the "two-key effect" for access to the mode. See Using Two Roles for Access to Unlimited Administration Mode, below, for details.

Limiting Role Access to Secret Templates

Anyone with access to modify your secret templates can change the definitions of the data being stored, and this access should be tightly controlled. Your secret templates are unlikely to need changing once you have defined them, so limiting access to a select number of individuals is typically sufficient.

Monitoring Roles with Event Subscriptions

Another option when protecting roles is to configure event subscriptions to notify appropriate staff in the event that Roles are changed or assigned. Event subscriptions are email alerts that can be sent to users, groups or specific email addresses, based on different events in Secret Server. There are also events available around the "unlimited administrator" role to further protect it from misuse.

Using Two Roles for Access to Unlimited Administration Mode

We recommend determining which role permissions should or should not be combined for users before assigning roles and allowing users access to the application. Part of that is planning access to the "unlimited administration" mode. Users with the "administer configuration unlimited admin" role permission can enable that mode. Once the system is in the mode, users with the "unlimited administrator" role permission can view all secrets in Secret Server and access all configuration settings. So a user with both permissions can enable the "unlimited administration" mode and then view all the secrets or make any configuration change.

To prevent a single person from having that much access, the two role permissions should be given to two different roles and only those roles, and nobody should have access to both of the roles. That enforces accountability and requires the cooperation of two people to enter "unlimited administration" mode.

A solution is to create the two roles, each containing one of the permissions, and then take those two permissions out of the day-to-day administrator role and any other roles besides the two. You can then assign either one of those roles to trusted people with no single person having both roles.

Thus, the access procedure is:

1. User A with the role with the "administer configuration unlimited admin" permission puts the system into "unlimited administration" mode. Not having the correct role, user A cannot make any changes requiring the "unlimited administrator" permission.
2. User B with the role with the "unlimited administrator" permission performs any configuration or accesses secrets only available to that role.
3. When User B is finished, user A takes the system out of "unlimited administration" mode.
4. User B can no longer make any changes requiring the "unlimited administrator" permission because roles with that permission can only be accessed in "unlimited administration" mode. User A cannot make any changes either because User A does not a the role with the "unlimited administrator" permission.

Additional safeguards included:

- Enabling or disabling "unlimited administration" mode is audited, and a comment should be provided each time it is enabled.
- When "unlimited administration" mode is enabled, a banner appears at the top of every Secret Server page notifying users that their secrets can currently be viewed by an unlimited administrator.
- Event subscription notifications should be set up to send an email to a specified user, group of users, or other email address whenever "unlimited administration" mode is enabled or disabled.
- All actions that are normally audited, such as secret views, edits, or permissions changes, are still audited while "unlimited administration" mode is enabled.

Encryption

DPAPI Encryption

Overview

The Data Protection API (DPAPI) is an option that provides an additional layer of security for the Secret Server encryption key. The Secret Server encryption key is contained within a file that is decrypted and used by the application to encrypt or decrypt the sensitive data that is stored in the Secret Server database. Using the DPAPI option in Secret Server ensures that the encryption key file is also encrypted with a key that only Windows knows and is only be usable on same server it was encrypted on. Anybody trying to configure Secret Server on another server using that DPAPI-encrypted key is blocked from doing so.

> The encryption key file, `encryption.config`, should be backed up and stored in a secure location before turning on DPAPI encryption. This allows you to restore a backup of the application on another server in a DR scenario. The file is in the Secret Server application directory.

> In order to encrypt the `encryption.config` file, the App Pool identity account needs permissions on the file.

Enabling and Disabling DPAPI

To turn on DPAPI encryption of the file, select **Configuration** from the **Admin** menu. Select the **Security** tab, click **Encrypt Key Using DPAPI**, and then type your password and acknowledge the warning before clicking **Confirm**. To decrypt the key, navigate to the same tab and click **Decrypt Key to not Use DPAPI**.

Using Clustering with DPAPI

You can use DPAPI while clustering is enabled for Secret Server, however there are a few things to take into consideration:

- Backup the encryption key before using this option, otherwise disaster recovery could prove impossible, should the server fail.
- You must initially transfer the un-encrypted key that DPAPI will encrypt to each Secret Server node. Secret Server
- You must enable DPAPI for Secret Server by accessing each server locally (browse to Secret Server while on the server it is installed on, and then enable DPAPI encryption).
- During upgrades, to avoid turning off DPAPI, you can copy all files over to secondary nodes *except* for `database.config` and `encryption.config`.

For more information about clustering Secret Server, see"Secret Server Clustering" on page 776

Protecting Your Encryption Key Using EFS

Encrypting File System (EFS) is a Microsoft technology that allows a user to encrypt files with their password. This means that only the user who encrypted the file will be able to access it, even if it is assigned to other users. If an administrator resets the password on this account and the account does not change its own password, then the file is not recoverable.

You can use EFS to protect your Secret Server encryption key. This allows only a single service account to access the file, and no other user can read the key unless they know the service account password. Below are the steps for encrypting your `encryption.config` and `database.config` files with EFS:

1. Backup your `encryption.config` and `database.config` files to a secure location. This is very important for DR recovery purposes.

   Important: This step is critical–If you lose access to your service account or the server fails, you will be unable to recover your secrets without these backup files.
2. Create a new service account or select an existing one. The service account should initially have privileges to log on a computer.
3. If you have already installed Secret Server and are using Windows authentication for database access, make sure the service account has access to the database.

4. Run the Secret Server application pool as this service account. See "Running the IIS Application Pool As a Service Account" on page 60 .
5. Give the service account full access to your Secret Server directory through Windows Explorer if it does not have it already.
6. Log on your server as the service account.
7. For both the `encryption.config` and `database.config files` (this instruction uses the former):
    a. Locate the `encryption.config` file in your Secret Server directory (usually `C:\inetpub\wwwroot\SecretServer`).
    b. Right-click the file and select **Properties**.
    c. Click the **General** tab.
    d. Click the **Advanced** button.
    e. Click to select the **Encrypt contents to secure data** check box.
    f. Click the **OK** button.
    g. Click the **Apply** button.
    h. If prompted, select the **Encrypt the file only** option.
    i. Click the **OK** button.
8. Log out of Windows and log back in as an administrator.
9. Confirm that the application still works by performing an IIS Reset (`IISReset` command at the command prompt) or recycling the application pool.
10. Ensure you can still log in and view your secrets.

SSL (TLS) and HSTS

We strongly recommend employing SSL (TLS) for Secret Server. Taking SSL a step further, Secret Server also supports HTTP Strict Transport Security (HSTS). HSTS is supported by modern browsers and tells the browser that a site is only accessible by SSL with a valid certificate, period. Even if there is a man-in-the-middle attack with a trusted, but different, SSL certificate, the browser will reject the SSL certificate. Consequently, this setting is very useful for protecting against forged SSL certificates or man-in-the-middle attacks.

For more information about configuring SSL certificates, see the . You can view additional information about HSTS in Securing with HTTP Strict Transport Security (HSTS).

SSH Key Validation

Host SSH Key verification is supported for use with heartbeat, proxied launchers, password changers, and discovery. Host SSH key verification can help ensure that the machine you are connecting to is a trusted host. Host SSH key verification will not pass credentials to the target machine unless the public key digest matches the SHA1 digest that Secret Server has on file. This helps prevent man-in-the-middle attacks.

Mapping an SHA1 Digest to Secrets

To configure host SSH key verification:

1. Navigate to **Secret Templates** from the **Admin** menu.
2. And add a field for the host's SSH key digest.
3. Click **Configure Extended Mappings**.
4. Add a **Server SSH Key** mapping to your newly created SSH key digest field.
5. On your secrets, add the SSH key digest of the hosts to your digest field. Verification takes effect the next time you connect to the host.

Validating SHA1 Digests for Unix Account Discovery

To validate SHA1 server digests for Unix account discovery, create a file named `KeyDigests.txt` in the root of the Secret Server website. Each line should contain an IP address or other computer identifier, a comma, and the SHA1 digest, for example:

```
192.168.1.5,7E:24:0D:E7:4F:B1:ED:08:FA:08:D3:80:63:F6:A6:A9:14:62:A8:15
apollo,7A:25:AB:38:3C:DD:32:D1:EA:86:6E:1C:A8:C8:37:8C:A6:48:F9:7B
```

When the file exists and has data, all scanned machines must match one of the SHA1 hashes in the file before scanning. Any computers that do not match will still show up on the "Discovery Network View" page, but authenticated scanning will not take place. That is, no credentials will be passed to the machine, and accounts will not be retrieved from the machine.

Disabling IIS HTTP Headers

Introduction

This section describes plugging some potential, minor but significant, information leaks by the Secret Server Web server. Web applications, such as Secret Server, may unintentionally disclose information about their underlying technologies through headers, error messages, version numbers, or other identifying information. An attacker can use that information to research vulnerabilities in those technologies to attack the application to breach the system.

Procedure

First, **hide the IIS version**. The HTTP header "X-Powered-By" reveals the version of IIS used on the server. To stop this, remove the header:

1. Open the IIS Manager.
2. In the **Connections** tree, select the website that Secret Server is running under.
3. Click the **HTTP Response Headers** button on the right. The HTTP Response Headers panel appears.
4. Click to select the **X-Powered-By** HTTP header.
5. Click the **Remove** button in the **Actions** panel. The header disappears.

Second, **hide the ASP.NET version**. The HTTP header "X-ASPNET-VERSION" reveals the version of ASP.NET being used by the Secret Server application pool. To stop this, remove the header:

1. Open the `web.config` file for Secret Server, which is located in the root directory for the website.
2. Inside the `<system.web>` tag, add the tag `<httpRuntime enableVersionHeader="false"/>`.
3. Save the file.

Third, **hide the server type**. The header line `Server: Microsoft-HTTPAPI/2.0` is added to the header by the .NET framework. To remove that information, you must update the Windows Registry:

> Do not simply remove the Server header variable–it will cause parts of Secret Server to malfunction.

1. Open the Windows Registry Editor.
2. Navigate to `Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\HTTP\Parameters`.
3. Change the `DisableServerHeader` (REG_DWORD type) registry key from `0` to `1`.

> There are other ways to hide the server type. We strongly recommend this one.

Adjusting CORS Policy Headers

By default, Secret Server only allows Cross-Origin Resource Sharing (CORS) to unauthenticated resources. CORS allows a restricted resource on a Web page to be requested from a different domain from that resource. To adjust this behavior, either hardening (such as blocking all CORS calls) or relaxing it, follow these instructions:

1. In the Secret Server installation folder, open the `web-appsettings.config` file in a text editor.
2. In the **appSettings** section add: `<add key="UseWebConfigCORS" value="true"></add>`
3. Save the file.
4. In the same folder, also open the `web.config` file in the text editor.
5. In the **system.webServer/httpProtocol/customHeaders** section, add: `<add name="Access-Control-Allow-Origin" value="[customer URL here]" /><add name="Access-Control-Allow-Headers" value="Content-Type" /><add name="Access-Control-Allow-Methods" value="GET, POST, PUT, DELETE, OPTIONS" />`
6. Save the file.

Additional Resources

- "Configuration Best Practices" on page 14
- Delinea.com

Accessing MS SQL Server with IWA

> This topic only applies to **Secret Server On-Premises**.

> Please see "Running the IIS Application Pool As a Service Account" on page 60 for additional information.

Introduction

Integrated Windows Authentication (IWA) requires:

- Installing a SQL Server instance
- Creating a new domain service account
- Granting access to SQL Server database
- Registering a service account to run IIS and ASP.NET
- Assigning an account as an application pool identity

Note: For instructions on Creating the SQL account or Installing SQL Server see "Installing and Configuring SQL Server" on page 99

Creating a Domain Service Account

The account needs access to the application server and database server. Ensure password expiration is not enabled or the account could lock you out of Secret Server.

Granting Access to SQL Server database

1. Connect to the Database instance using SQL Management Studio.
2. Right click on the Security node (ensure this is the top most security node under the instance and not under the database name itself) and select **New > Login**.
3. Enter the Login name as `Domain\Username`.
4. Ensure **Windows Authentication** radio button is selected.
5. If you have already created the database, then under **User Mappings** select the database and grant dbOwner permission. Otherwise, if you plan to have the Database created for you, under **Server Roles** select dbCreator.
6. Click the **Ok** button.

Assigning Account as Identity of Application Pool

1. Open IIS (Run command inetmgr).
2. Click the Application Pool node.
3. Select Secret Server's Application Pool (default is SecretServerAppPool).
4. On the Right panel, Click .
5. Scroll down to the **Identity** row under **Process Model**.
6. In the popup, select **Custom Account > Set**.
7. Type the user as domain\username.
8. Type the password.
9. Click the **Ok** button.
10. Recycle the application pool by clicking the **Recycle..** button under the **Application Pool** tasks.

Considerations for an Externally Accessible Secret Server

This topic only applies to **Secret Server On-Premises**.

Secret Server can be hosted externally, like any other IIS website. We recommend using all security measures for Secret Server that you would use for any server directly accessible to the internet. We also recommend the measures described below.

Limiting the Attack Surface

- The Secret Server application should reside on a dedicated server in a DMZ.
- Secret Server and its database should reside on separate servers. If a hole for SQL connections can be opened in the DMZ firewall, the database can reside on the other side of the DMZ firewall.

Using Secure Connections

- Use HTTPS to access to the website.
- Use SSL to connect to the Secret Server database.
- Use LDAPS to connect the web server to Active Directory.

Setting Up Remote Password Changing

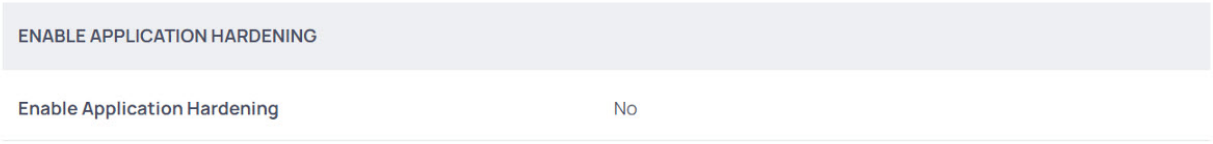
By default, Secret Server changes passwords on devices and accounts directly from the web server where it is installed, but when Secret Server is installed in a DMZ zone, it does not have direct network connections to these devices and accounts.

However, you can enable Secret Server to change passwords throughout your network over a specified port using distributed engines. See "Distributed Engine Overview" on page 723 for more information on setting up and using Secret Server Distributed Engines.

Enabling Application Hardening

This topic only applies to **Secret Server On-Premises**.

Application Hardening can be found in the **Admin > Configuration > Security tab**.



When this feature is enabled, extra checks are done to make sure user records have not been tampered with. If a user's record has been found to have been modified by someone other than Secret Server itself, they will not be able to log in. New Event Subscriptions have also been added to send alerts if tampering is detected. This feature is only available in Secret Server on-premise.

This feature is a configurable setting designed to modify the system so that a database administrator cannot effectively modify or create a User Record through the database directly–this action needs to happen in the application itself. This includes:

- Resetting a password
- Disabling or changing Two-Factor Authentication
- Creating a brand-new record with a known password

Hardening RDS Hosts for Session Connector

This topic only applies to **Secret Server On-Premises**.

Overview

Delinea Secret Server offers a variety of session launching methods, including the use of Microsoft Remote Desktop Services (RDS) in conjunction with Secret Server Session Connector. This setup allows users to initiate privileged sessions via an RDS host without needing additional installations on client devices. All that's required is a Remote Desktop Protocol (RDP) client capable of opening RDP files furnished by Secret Server. Given the integral role of RDS hosts in this configuration, it is essential to harden them.

Prerequisites

Complete and validate the installation, setup, and functionality of Secret Server Session Connector before proceeding with the hardening process. If the RDS host is already in production, ensure a full backup of the RDS host and its configurations are available.

The Issue

RDS hosts offer access to a published application. Even though users will connect with unique, non-privileged, and randomly generated credentials on the RDS host, they can still interact extensively with the OS. For example, using the <CTR>-<ALT>-<END> command in the published application accesses the task manager on the RDS host, letting the user initiate additional programs and view the host's local file system. This could potentially allow the user to start additional unauthorized applications.

The Solution

Limit user access, reduce the attack surface, and ensure secure session handling through a combination of OS-level settings, GPOs, and application configurations.

Operating System Hardening

- **Patching**: Regularly update the OS to patch known vulnerabilities.
- **Unnecessary Services**: Disable services and applications that are not required for the server's role.
- **Antivirus/Antimalware**: Install and regularly update antivirus software.
- **Limit User Access**: Only grant least-privilege access rights.

Network Hardening

- **Firewalls**: Ensure a firewall is in place to filter unwanted inbound and outbound traffic. Only allow the necessary ports (for example, 3389 for RDP).
- **Virtual Private Network (VPN):** Avoid directly exposing the RDS server to the internet. Employ a VPN to facilitate secure remote connectivity.
- **Strategic Network Segmentation:** Position the RDS server within a distinct VLAN, ensuring optimal traffic isolation and mitigating risks associated with broader network vulnerabilities.
- **Intrusion Detection:** Deploy an Intrusion Detection System (IDS) to continuously oversee and promptly address any unusual or potentially harmful network behavior.

Local User Hardening via PowerShell Script

For these local user adjustments, you can use the provided PowerShell script. The following settings are deployed into the default user profile via the PowerShell script:

- **Hide Local Drives:** Prevent users from viewing the RDS host's local drives.
- **Lock CTRL-ALT-DEL Screen:** Restrict users from locking the system, altering passwords, or launching the task manager.
- **Remove Shutdown Button:** Although standard users lack the permission to shut down the RDS host, it is best to hide the shutdown buttons to avoid confusion.
- **Disable Screensaver and Lock Screen:** Users do not know the password for the randomly generated user. Configure the system to prevent screensaver activation or session lockouts.
- **Lock Internet Explorer Settings:** Even if Internet Explorer is not in active use, configure its settings to enhance security.

PowerShell Script

Overview

The script performs the following actions:

- Sets the execution policy to RemoteSigned.
- Creates a backup of the default user's `ntuser.dat` file.
- Loads the default user profile registry hive for editing.
- Hides local drives of the RDS Server.
- Disables local drive viewing.
- Adjusts <CTRL>-<ALT>-<DEL> options.
- Disables Task Manager, Change Password, and Lock.
- Disables Internet Explorer context menu.
- Disables Screensaver and Lock Screen.
- Disables Developer mode on Internet Explorer.

- Removes shutdown buttons.
- Blocks access to Control Panel and PC Settings.
- Disables Windows Store.
- Checks and verifies the changes made to the registry settings.
- Unloads the default user profile registry hive after edit.

Download

Access the script via the box below.

Deployment

To ensure smooth execution of this PowerShell script:

- Run the script before applying hardening group policies to the RDS host.
- Right click and select **Run as administrator**.

By default, the script will only hides C and D drives. If you wish to hide additional drives, please change the prospective registry values in the following sections prior to running the script:

> See How-to: Hide drive letters from Windows Explorer for details on calculating and setting the desired values.

```
# Hide Drive Setting
```

Current value is 12 (C + D)

```
# No View Drive Setting
```

Current value is 12 (C + D)

The script is designed to provide visual feedback on the status of each setting:

- Green: Denotes settings that are correctly configured and aligned with security and operational standards.
- Red: Denotes settings that are not found within the system registry.
- Yellow: Denotes settings that are present but have a value mismatch, meaning the current configuration does not align with the expected standard.

```
<#
.SYNOPSIS
This script is designed to harden the default user profile on a system.
.DESCRIPTION
This script performs the following actions:
- Sets the execution policy to RemoteSigned
- Creates a backup of the default user's ntuser.dat file.
- Loads the default user profile registry hive for editing.
- Hides local drives of the RDS Server.
- Disables viewing of local drives.
- Adjusts CTRL-ALT-DEL options.
```

```
- Disables Task Manager, Change Password, and Lock.
- Disables Internet Explorer context menu.
- Disables Screensaver and Lock Screen.
- Disables Developer mode on Internet Explorer.
- Removes Shutdown Buttons.
- Blocks access to Control Panel and PC Settings.
- Disables Windows Store.
- Checks and verifies the changes made to the registry settings.
- Unloads the default user profile registry hive after editing.
- Reverts the execution policy back to its original state.
Note: This script needs to be run as an administrator.
#>
# User Confirmation
$confirmation = Read-Host "This script will make changes to system settings. Do you
want to continue? (Y/N)"
if ($confirmation -ne 'Y') {
    Write-Host "Script execution cancelled by user."
    exit
}
# Check if the script is run as administrator
if (-not ([Security.Principal.WindowsPrincipal]
[Security.Principal.WindowsIdentity]::GetCurrent()).IsInRole
([Security.Principal.WindowsBuiltInRole] "Administrator")) {
    Write-Host "Please run this script as an administrator." -ForegroundColor Red
    exit
}
# Store the current execution policy
$originalPolicy = Get-ExecutionPolicy
Write-Host "Storing the current execution policy: $originalPolicy"
# Set execution policy to RemoteSigned
Set-ExecutionPolicy RemoteSigned -Force
Write-Host "Execution policy set to RemoteSigned"
# Create a backup of ntuser.dat to the current directory with a timestamp
$backupFileName = ".\ntuser.dat-backup-$(Get-Date -Format 'yyyy_MM_dd_HH_mm_ss')"
Copy-Item -Path 'C:\users\default\ntuser.dat' -Destination $backupFileName
Write-Host "Backup of ntuser.dat created: $backupFileName"
# Load the default user profile registry hive
$defaultProfileRegistry = 'C:\users\default\NTUSER.DAT'
$registryHive = 'HKEY_USERS\RDSProfile'
$null = reg load $registryHive $defaultProfileRegistry
Write-Host "Default user profile registry hive loaded."
$key = "Registry::$registryHive"
# Function to ensure a registry key exists
function Ensure-RegistryKey {
    param(
        [string]$Path
    )
    if (-not (Test-Path $Path)) {
        New-Item -Path $Path -Force | Out-Null
        Write-Host "Registry key created: $Path"
    }
}
```

```
# Function to check a registry key value
function Check-RegistrySetting {
    param (
        [string]$Path,
        [string]$Name,
        [string]$Type,
        $ExpectedValue
    )
    $value = Get-ItemPropertyValue -Path $Path -Name $Name -ErrorAction
SilentlyContinue
    if ($null -eq $value) {
        Write-Host "Setting NOT FOUND: $Path\$Name" -ForegroundColor Red
        return
    }
    if ($Type -eq 'DWORD' -and $value -eq $ExpectedValue) {
        Write-Host "Setting OK: $Path\$Name = $value" -ForegroundColor Green
    }
    elseif ($Type -eq 'String' -and $value -eq $ExpectedValue) {
        Write-Host "Setting OK: $Path\$Name = $value" -ForegroundColor Green
    }
    else {
        Write-Host "Setting MISMATCH: $Path\$Name = $value (Expected: $ExpectedValue)"
-ForegroundColor Yellow
    }
}
# Hide local drives (Calculate required value using:)
$hideDriveKey = "$key\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer"
Ensure-RegistryKey -Path $hideDriveKey
New-ItemProperty -Path $hideDriveKey -Name 'NoDrives' -PropertyType DWORD -Value 12 -
Force
# No view on local drives (Calculate required value using:)
$viewDriveKey = "$key\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer"
Ensure-RegistryKey -Path $viewDriveKey
New-ItemProperty -Path $viewDriveKey -Name 'NoViewOnDrive' -PropertyType DWORD -Value
12 -Force
# Adjust CTRL-ALT-DEL options
$ctrlAltDelKey = "$key\Software\Microsoft\Windows\CurrentVersion\Policies\System"
Ensure-RegistryKey -Path $ctrlAltDelKey
New-ItemProperty -Path $ctrlAltDelKey -Name 'DisableChangePassword' -PropertyType DWORD
-Value 1 -Force
New-ItemProperty -Path $ctrlAltDelKey -Name 'DisableLockWorkstation' -PropertyType
DWORD -Value 1 -Force
New-ItemProperty -Path $ctrlAltDelKey -Name 'DisableTaskMgr' -PropertyType DWORD -Value
1 -Force
# Remove Shutdown Buttons
$shutdownKey = "$key\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer"
Ensure-RegistryKey -Path $shutdownKey
New-ItemProperty -Path $shutdownKey -Name 'NoClose' -PropertyType DWORD -Value 1 -Force
# Disable Screensaver / Lockscreen
$screensaverKey = "$key\Software\Policies\Microsoft\Windows\Control Panel\Desktop"
Ensure-RegistryKey -Path $screensaverKey
```

```
New-ItemProperty -Path $screensaverKey -Name 'ScreenSaveActive' -PropertyType String -
Value '0' -Force
New-ItemProperty -Path $screensaverKey -Name 'ScreenSaverIsSecure' -PropertyType String
-Value '0' -Force
# Disable Internet Explorer Context Menu
$ieContextKey = "$key\Software\Policies\Microsoft\Internet Explorer\Restrictions"
Ensure-RegistryKey -Path $ieContextKey
New-ItemProperty -Path $ieContextKey -Name 'NoBrowserContextMenu' -PropertyType DWORD -
Value 1 -Force
# Disable Internet Explorer Developer Tools
$ieDevToolsKey = "$key\Software\Policies\Microsoft\Internet Explorer\IEDevTools"
Ensure-RegistryKey -Path $ieDevToolsKey
New-ItemProperty -Path $ieDevToolsKey -Name 'Disabled' -PropertyType DWORD -Value 1 -
Force
# Disable Internet Explorer First Run
$ieFirstRunKey = "$key\Software\Policies\Microsoft\Internet Explorer\Main"
Ensure-RegistryKey -Path $ieFirstRunKey
New-ItemProperty -Path $ieFirstRunKey -Name 'DisableFirstRunCustomize' -PropertyType
DWORD -Value 2 -Force
# Disable Internet Explorer Settings Tabs
$iePanelKey = "$key\Software\Policies\Microsoft\Internet Explorer\Control Panel"
Ensure-RegistryKey -Path $iePanelKey
New-ItemProperty -Path $iePanelKey -Name 'SecurityTab' -PropertyType DWORD -Value 1 -
Force
# Block access to Control Panel and PC Settings
$controlPanelKey = "$key\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer"
Ensure-RegistryKey -Path $controlPanelKey
New-ItemProperty -Path $controlPanelKey -Name 'NoControlPanel' -PropertyType DWORD -
Value 1 -Force
# Disable Windows Store
$storeKey = "$key\Software\Policies\Microsoft\WindowsStore"
Ensure-RegistryKey -Path $storeKey
New-ItemProperty -Path $storeKey -Name 'RemoveWindowsStore' -PropertyType DWORD -Value
1 -Force
# Check local drives are hidden
Check-RegistrySetting -Path
"$key\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer" -Name 'NoDrives' -
Type DWORD -ExpectedValue 12
# Check viewing of local drives is disabled
Check-RegistrySetting -Path
"$key\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer" -Name 'NoViewOnDrive'
-Type DWORD -ExpectedValue 12
# Check CTRL-ALT-DEL options
Check-RegistrySetting -Path
"$key\Software\Microsoft\Windows\CurrentVersion\Policies\System" -Name
'DisableChangePassword' -Type DWORD -ExpectedValue 1
Check-RegistrySetting -Path
"$key\Software\Microsoft\Windows\CurrentVersion\Policies\System" -Name
'DisableLockWorkstation' -Type DWORD -ExpectedValue 1
Check-RegistrySetting -Path
"$key\Software\Microsoft\Windows\CurrentVersion\Policies\System" -Name 'DisableTaskMgr'
-Type DWORD -ExpectedValue 1
# Check shutdown buttons are removed
```

```
Check-RegistrySetting -Path
"$key\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer" -Name 'NoClose' -Type
DWORD -ExpectedValue 1
# Check screensaver and lock screen are disabled
Check-RegistrySetting -Path "$key\Software\Policies\Microsoft\Windows\Control
Panel\Desktop" -Name 'ScreenSaveActive' -Type String -ExpectedValue '0'
Check-RegistrySetting -Path "$key\Software\Policies\Microsoft\Windows\Control
Panel\Desktop" -Name 'ScreenSaverIsSecure' -Type String -ExpectedValue '0'
# Check Internet Explorer context menu is disabled
Check-RegistrySetting -Path "$key\Software\Policies\Microsoft\Internet
Explorer\Restrictions" -Name 'NoBrowserContextMenu' -Type DWORD -ExpectedValue 1
# Check Developer mode on Internet Explorer is disabled
Check-RegistrySetting -Path "$key\Software\Policies\Microsoft\Internet
Explorer\IEDevTools" -Name 'Disabled' -Type DWORD -ExpectedValue 1
# Check Control Panel and PC Settings access is blocked
Check-RegistrySetting -Path
"$key\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer" -Name
'NoControlPanel' -Type DWORD -ExpectedValue 1
# Check Windows Store is disabled
Check-RegistrySetting -Path "$key\Software\Policies\Microsoft\WindowsStore" -Name
'RemoveWindowsStore' -Type DWORD -ExpectedValue 1
# Unload the registry hive
$null = reg unload $registryHive
Write-Host "Registry hive unloaded."
# Restore the original execution policy
Set-ExecutionPolicy $originalPolicy -Force
Write-Host "Execution policy restored to $originalPolicy."
Write-Host "Script execution completed. Please review the settings above. For any
discrepancies, consult the log file and consider manual verification." -ForegroundColor
Green
```

Machine Hardening via Group Policy Objects (GPO)

The CIS (Center for Internet Security) Benchmark is an invaluable tool, offering a solid foundation for system configurations. Serving a wide range of entities from commercial sectors to government agencies, it provides detailed guidelines, ensuring robust system hardening. Similarly, the STIG (Security Technical Implementation Guide) framework is revered for its comprehensive approach, especially tailored to fortify systems entrusted with national security and classified data.

Integrating Our Custom Hardening GPO

Overview

Our custom hardening GPO can be easily integrated into your Active Directory environment. While the policy comes with a default configuration based on our insights and some industry best practices, you can modify it to better fit your organization's specific requirements.

> If you are currently implementing STIG policies, we advise against integrating our GPO. The overlap and potential conflicts may compromise system integrity and result in a decreased STIG compliance score. For detailed guidance on this, please refer to the relevant section on “STIG Compliance with Session Connector Functionality”.

Adding the Custom GPO to Your AD Environment

Download

Download the Hardening RDS Hosts for Session Connector .

Deploying the GPO File

> We recommend applying the GPO exclusively to the RDS host being used in conjunction with the session connector.

1. Extract the contents of the zip file to your desired location
2. Open the Group Policy Management Console (GPMC).
3. Create a new GPO.
4. Right click the new GPO and select **Import Settings**. An Import Settings Wizard appears.
5. Click the **Next >** button to skip backing up the GPO.
6. Populate the **Backup folder** text box on the **Backup location** page with the location you unzipped the file to, and click the **Next >** button.
7. On the **Source GPO** page click the GPO you just added in the **Backed up GPOs** list, and click the **Next >** button.
8. On the **Migrating References** page, click to select the **Copying them identically from the source** selection button, and click the **Next >** button.
9. On the **Completing the Import Settings Wizard** page, click the **Finish** button.

Applying Custom GPOs to a Chrome Browser

This section describes how to configure Google Chrome ADM/ADMX templates to use custom GPOs:

1. Download the Google Chrome Enterprise Browser. This includes the ADM/ADMX templates and Google updater ADMX template update.
2. ADMX Deployment:

   For a central store deployment, place the `.admx` files here:

   `\\<domain>\SYSVOL\<domain>\Policies\PolicyDefinitions`

   Without a central store, place the `.admx` files here:

   `%systemroot%\PolicyDefinitions`
3. ADML Deployment:

   For a central store deployment, place the `.adml` files here (adjusting the language tag):

   `\\<domain>\SYSVOL\<domain>\Policies\PolicyDefinitions\en-US`

   Without a central store, place the `.adml` files here (adjusting the language tag):

   `%systemroot%\PolicyDefinitions\en-US`

As a best practice, always pilot any new GPO within a controlled setting prior to a full-scale deployment on your production systems.

STIG Compliance with Session Connector Functionality

Overview

While the Security Technical Implementation Guide (STIG) provides a robust framework for system security, some functional requirements, such as those of the session connector for RDP sessions, may necessitate deviations from STIG policies that might marginally lower your STIG compliance score.

The following policy configurations optimize session connector functionality on your RDS host. These recommendations offer a balance between operational necessities and security best practices. If you are employing the session connector for RDP, ensure that these policies are applied across your target systems as well.

User Rights Assignments

Configuration Path

Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > User Rights Assignments

Policies

Allow local login:

- Users
- Administrators

Allow login through terminal services:

- Remote Desktop Users
- Administrators

Deny login through terminal services:

- Guests
- <NETBIOS>\Domain Admins
- <NETBIOS>\Enterprise Admins

Remote Desktop Session Host Security Configurations

Configuration Path

Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Security

Policies

Always prompt for password upon connection: disabled

It is imperative to validate these configurations in a controlled environment before deploying them in production, ensuring both compliance and operational integrity. Also, coordinate with IT governance or compliance teams to ensure alignment with organizational guidelines.

Application Lockdown

- An application white listing strategy is instrumental in elevating the security posture of an organization.
- By permitting only approved applications to execute, the risk of malicious software execution and potential security breaches is significantly reduced.
- Consider tools such as Microsoft AppLocker for application control solutions.

Application Hardening via AppLocker

You can easily integrate our custom AppLocker GPOs into your Active Directory environment. While these policies come bundled with two different AppLocker configuration files based on our insights, you can choose the one that best aligns with your objectives and modify it as needed.

Session Connector Considerations

For the seamless operation of Session Connector launchers, the following executables must be explicitly allowed to run within your policies:

- rdpwin.exe
- rdpwin.bootstrapper.exe
- rdpwin.rdpclient.exe
- rdpwin.watchdog.exe

These executables are crucial for the proper functioning of session connector launchers as they perform various roles from launching the applications and managing RDP client sessions, to monitoring the health of the launcher. Ensuring these are allowed to run is vital for maintaining operational integrity and ensuring users can initiate and maintain their sessions without interruption.

AppLocker Policies

Both policies target the “Remote Desktop Users” group on the RDS host and will apply deny rules to prevent these users from accessing software. When launching an application, the Session Connector will grant logon rights by creating an ephemeral local user and adding this user to the “Remote Desktop Users” group.

During the installation of Remote Desktop Services (RDS), if you elected to associate the session collection with the default “Domain Users” group, the restrictions of these policies will apply to **all** domain users. RDS nests the “Domain Users” security group within the “Remote Desktop Users” group.

This integration is critical if you plan to run an application in the context of a specific domain user and wish for the AppLocker policies to be enforced.

If “Domain Users” is nested within “Remote Desktop Users,”, we recommend assigning your administrative user a new primary group in Active Directory Users & Computers (ADUC) and remove the administrative user from the “Domain Users” security group. This allows your administrative user to perform administrative tasks on the RDS host.

StrictAppLockerPolicy.xml

This policy restricts members of the “Remote Desktop Users” group from launching all applications, scripts, windows installers, files, and packaged apps, regardless of their path, unless explicitly defined in the exceptions.

Here is a list of the exceptions, which are primarily required for basic session connector functionality:

If you elect to use this policy, you should manually edit the AppLocker rule "Deny All (Remote Desktop Users)" and explicitly add exceptions here for your approved applications.

- conhost.exe: Handles console windows, providing the user interface for console applications.
- csrss.exe: Critical for system functionality, handling console windows, and the shutdown process.
- dllhost.exe: Hosts DLLs on behalf of other applications, potentially for COM components or other DLLs.
- dwm.exe: Manages graphical effects in the user interface.
- explorer.exe: The main shell of Windows, providing the desktop environment, taskbar, and Start menu.
- lsass.exe: Handles authentication and password policies, managing user login.
- osk.exe: Provides an on-screen keyboard for accessibility.
- rdpclip.exe, rdpinit.exe: Allows for clipboard sharing between the local machine and the RDP session.
- rdpinit.exe: Initializes an RDP session.
- rdpsa.exe, rdpsaproxy.exe: System executables that support the proper functionality of Remote Desktop Services.
- rdpshell.exe: Provides a shell or user interface for the RDP session.
- runonce.exe: Runs commands or scripts when a user logs on to the system for the first time after a restart.
- screenmagnifier.exe: Accessibility feature for screen magnification.
- sethc.exe: Related to Sticky Keys, a part of accessibility features.
- sihost.exe: Shows various user interface elements, such as system information on the secure desktop.
- smss.exe: Initializes the user session during Windows startup.
- svchost.exe: Hosts various Windows services.
- taskhostw.exe: A generic host process for running DLL-based services.
- tsthemes.exe: Related to RDP theme and appearance settings.
- userinit.exe: Initializes user settings and launches the user shell after log on.
- wermgr.exe: Involved in error reporting and solutions.
- winlogon.exe: Handles the login and logout procedures.

- rdpwin.exe, rdpwin.bootstrapper.exe, rdpwin.rdpclient.exe, rdpwin.watchdog.exe: Delinea files required for session connector functionality.

DefaultAppLockerPolicy.xml

This is a less restrictive policy and only restricts members of the “Remote Desktop Users” group from launching the following applications within the Windows directory:

> If you elect to use this policy you should manually edit the AppLocker Executable Rules and explicitly add or remove denied applications here based on your organization’s requirements.

- at.exe: Schedules commands at a specific time.
- bcdedit.exe: Manages Boot Configuration Data.
- bitsadmin.exe: Creates download or upload jobs, facilitating unauthorized data transfer or malware delivery.
- cacls.exe: Changes file and directory permissions, which could modify system security settings.
- certreq.exe: Performs various certification authority (CA) certificate functions.
- certutil.exe: Manages CA certificates.
- cipher.exe: Alters the encryption of directories and files on NTFS partitions.
- cmd.exe: Allows arbitrary command execution.
- cmstp.exe: Installs or removes Connection Manager service profiles, which can be misused for arbitrary code execution.
- compmgmtlauncher.exe: Launches the Computer Management console.
- control.exe: Accesses Control Panel items.
- cscript.exe/wscript.exe: Executes VBScript or JScript.
- csvde.exe: Imports and exports Active Directory data in a comma-separated format.
- dcomcnfg.exe: Manages DCOM settings. Unauthorized modifications can lead to unauthorized remote control or data access.
- dism.exe: Services Windows images and could be misused to alter them.
- diskpart.exe: Manages disk partitions.
- dnscmd.exe: Manages DNS servers from the command line, which can be misused to change DNS configurations.
- driverquery.exe: Lists installed device drivers, which can be used for system reconnaissance.
- dsdbutil.exe: A command-line tool that can be used to manage Active Directory databases.
- dsquery.exe: Searches for objects in the directory from the command line.
- eventcreate.exe: Creates custom events in specified event logs.
- eventvwr.exe: Views system logs, which can be exploited for "bait and switch" attacks.
- expand.exe: Expands compressed files, which can be misused for bypassing software restriction policies.
- extrac32.exe: Extracts .cab files from the command line, similar to expand.exe.

- findstr.exe: Searches for strings in files, which can be used for data reconnaissance.
- finger.exe: Displays information about remote users on older systems.
- forfiles.exe: Executes commands on a set of files.
- fsutil.exe: Manages FAT and NTFS file systems, which could tamper with file systems.
- ftp.exe: FTP client.
- getmac.exe: Displays MAC addresses of network adapters.
- gpresult.exe: Displays group policy information for remote users.
- infdefaultinstall.exe: Installs driver packages.
- klist.exe: Manages Kerberos tickets, misuse of which can lead to unauthorized access.
- lodctr.exe: Updates performance counter-related registry values, which can be misused.
- makecab.exe: Creates .cab files, potentially misused for file obfuscation.
- mavinject.exe: Injects DLLs, which can be misused for privilege escalation.
- mmc.exe: Microsoft Management Console, which hosts various management snap-ins.
- msconfig.exe: Modifies system startup items.
- mshta.exe: Executes HTML applications, which can be misused for unauthorized script or payload execution.
- msiexec.exe: Installs software packages.
- msinfo32.exe: Retrieves detailed system information.
- net.exe: Manages users, groups, and more.
- netsh.exe: Modifies network configurations.
- nltest.exe: Provides network diagnostics, which can be used for reconnaissance.
- odbcad32.exe: Manages database connections.
- openfiles.exe: Displays files opened by remote users, which can be used for reconnaissance.
- pathping.exe: A network diagnostic tool, which is useful for reconnaissance.
- perfmon.exe: Monitors system performance.
- powershell.exe: PowerShell, a powerful scripting environment.
- printbrmui.exe: Manages printer migrations.
- query.exe: Provides information on user sessions, processes, and more, which is useful for reconnaissance.
- quser.exe: Displays information about logged-on users, which can be used for reconnaissance.
- rasdial.exe: Manages network connections, which can be misused to establish unauthorized connections.
- reg.exe: Edits the registry from the command line.
- regedit.exe, regedt32.exe: Modifies the Windows registry.
- robocopy.exe: A file copy tool, which can be misused to transfer data.
- schtasks.exe: Manages scheduled tasks.

- sc.exe: Manages services.
- secedit.exe: Configures and analyzes system security by comparing it to a template.
- shutdown.exe: Shuts down or restarts the system.
- sdbinst.exe: Installs custom database files, which can be misused for unauthorized binary execution.
- sysprep.exe: Prepares systems for cloning, which can remove security settings.
- takeown.exe: Regains file access by taking ownership, which can be misused to modify system files.
- taskmgr.exe: Manages processes and views performance stats.
- telnet.exe: Telnet client.
- tzutil.exe: Time zone utility, which can be misused in specific contexts.
- vssadmin.exe: Manages volume shadow copies.
- wevtutil.exe: Manages event logs, which can be misused to clear them.
- werfault.exe: Windows Error Reporting tool, which has been misused for malicious intent in some scenarios.
- whoami.exe: Displays information about the current user.
- winrs.exe: Runs commands on remote computers.
- wmic.exe: Windows Management Instrumentation command line, which can be used for reconnaissance or administration.
- wsmprovhost.exe: Hosts Windows Remote Management, which can be misused in specific contexts.
- wusa.exe: Installs Windows updates, which can be misused to install malicious updates.
- xwizard.exe: Extensible Wizard Framework, which has been used in UAC bypass techniques.

Integrating Our Custom AppLocker Policies

These steps enhance security by restricting the execution of potentially harmful or unauthorized applications.

Download

Access the GPO and AppLocker policies via the Hardening RDS Hosts for Session Connector

Deployment

> We recommend applying this GPO exclusively to the RDS host being used in conjunction with session connector.

1. Extract contents of the zip file to your desired location.
2. Follow “Guidelines for Machine Hardening via GPO” for instructions on how to import a GPO.
3. Edit the new GPO and import the desired AppLocker policy:
   a. Navigate to the following path in the GPO editor: Computer Configuration > Policies > Windows Settings > Security Settings > Application Control Policies > AppLocker.

b. Right click and select **Import Policy**

c. Select the desired policy.

d. Click the **Open** button. An Import Policy dialog box appears.

e. Click the **Yes** button. An AppLocker dialog box appears.

f. Click the **OK** button. The dialog box disappears.

g. Ensure the new rules appear in the GPO editor.

AppLocker Diagnostic Procedures

Running in Audit Only Mode

If you experience issues with applications executing, we recommend switching AppLocker's operational mode from "Enforce rules" to "Audit only." This adjustment enables a comprehensive audit while ensuring uninterrupted application functionality. To switch the mode:

1. Navigate to the following path in the AppLocker GPO: Computer Configuration > Policies > Windows Settings > Security Settings > Application Control Policies > AppLocker.
2. Right click and select **Properties**
3. Set the **Executable rules** dropdown list to **Audit only**.

Using AppLocker Audit Logs

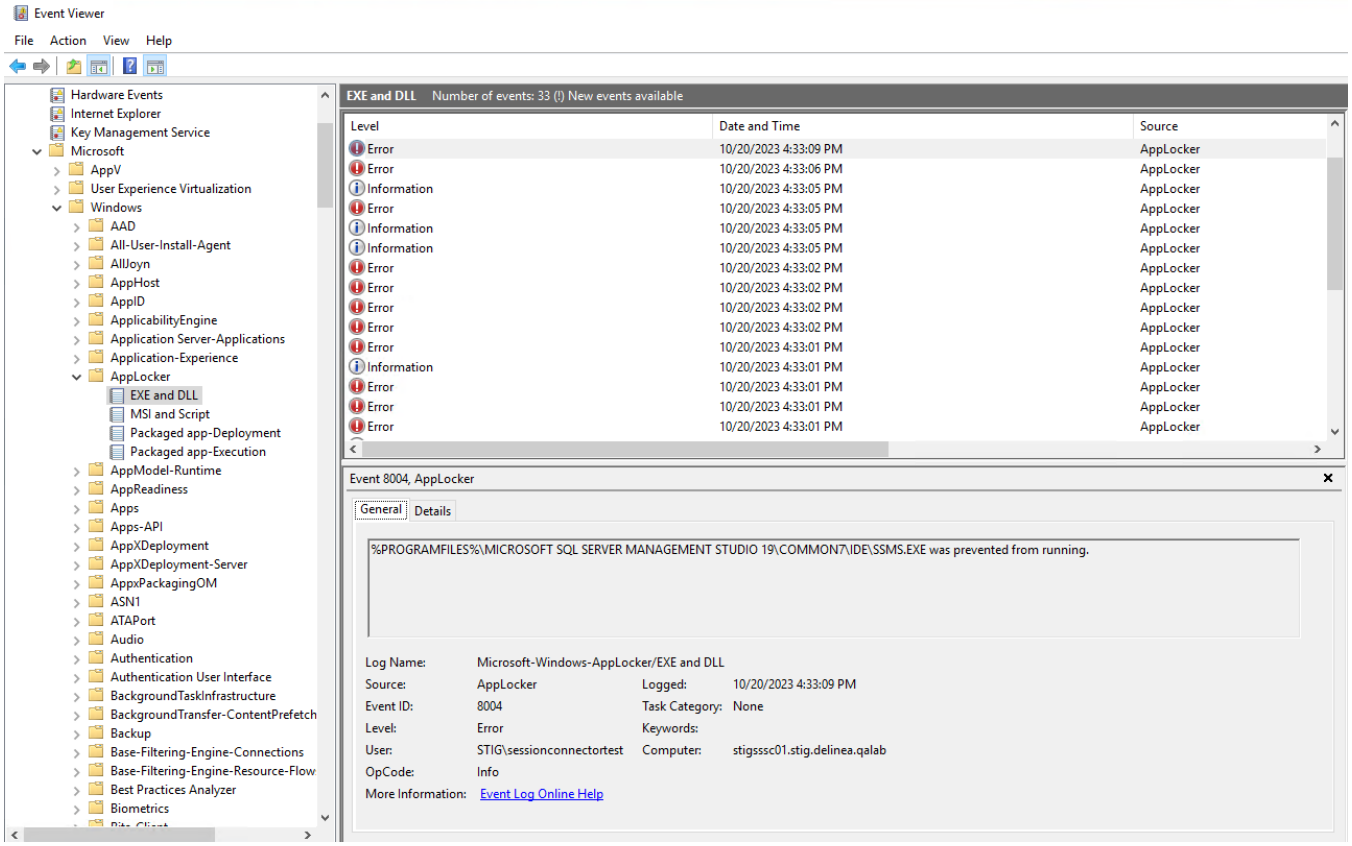
Procedure

The AppLocker audit logs provide invaluable insights and are instrumental for troubleshooting:

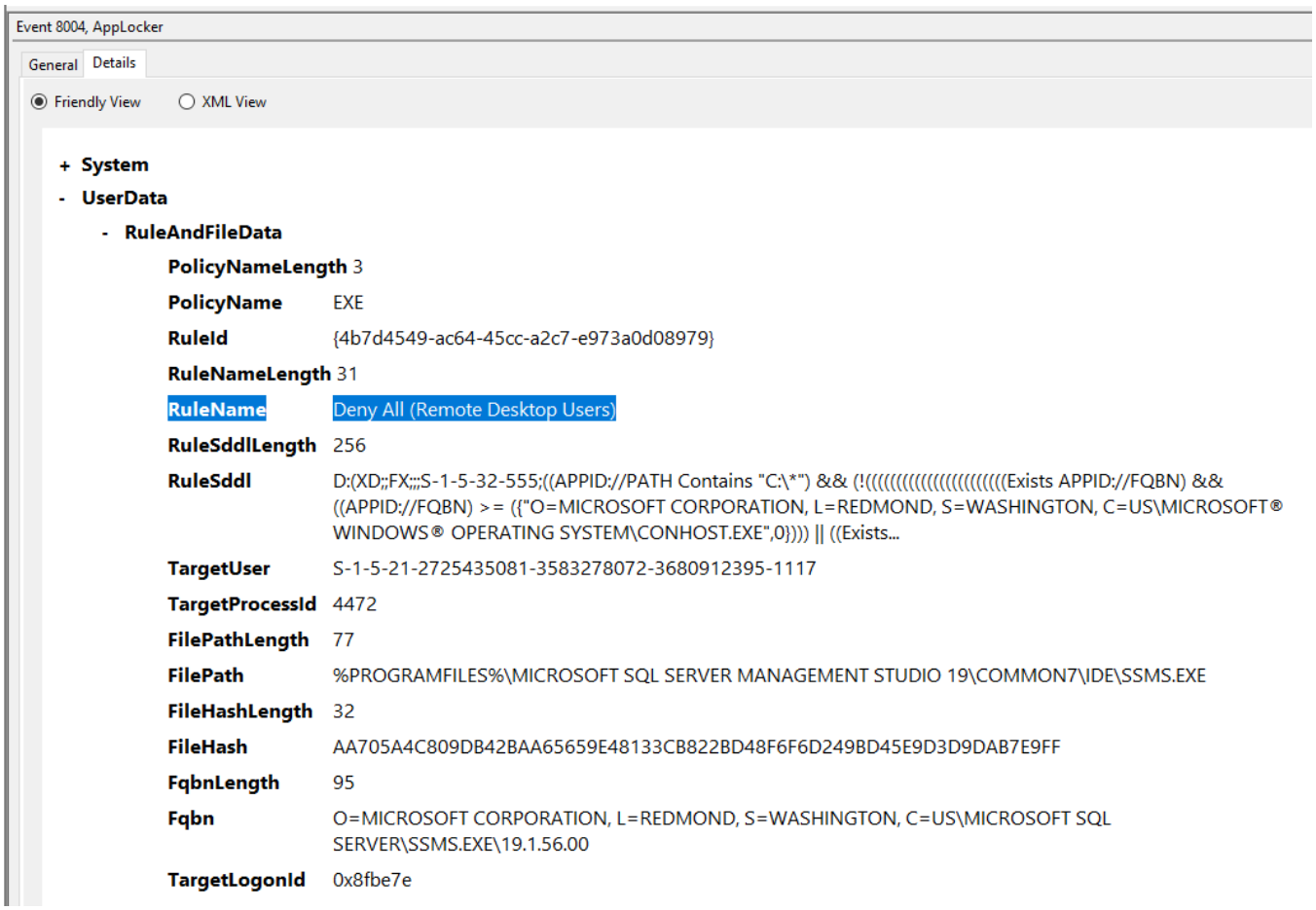
1. **Access Event Viewer**: Press Windows Key + R, type `eventvwr.msc`, and press **<Enter>**.
2. **Navigate to AppLocker Logs**: Go to Applications and Services Logs > Microsoft > Windows > AppLocker.
3. **Select Log Files**: Click on the log categories such as EXE and DLL to view events.
4. **Filter Audit Events**: Use **Filter Current Log** on the right panel, and specify event levels or IDs to isolate audit results.
5. **Analyze Event Details**: Select events and review the **General** and **Details** tabs for information on the audited application and actions taken.
6. **Adjust Policies**: Analyze to pinpoint issues, and modify AppLocker policies as needed for resolution.

Example

SSMS.EXE will not run:



Click the details tab to display additional information regarding the rule preventing the application from launching:



Additional Suggestions

Monitoring and Logging

We recommend:

- Employing proactive system and application log analysis to detect irregularities and potential threats.
- Setting up automated alerts for critical events, notably patterns like successive failed login attempts, to ensure timely interventions.

Continuous Maintenance and Oversight

We recommend:

- Remaining informed and promptly acting on any updates or advisories from Delinea.
- Routinely assessing and refining security configurations to ensure they align with evolving best practices and organizational needs.

> Remember, while hardening is essential, it is equally crucial to maintain usability and functionality for end-users. We strongly recommend thoroughly testing any security configurations in a staging environment before deploying to production.

Hardware Security Module Overview

> This topic only applies to **Secret Server On-Premises**.

Hardware Security Modules (HSMs) are specialized hardware devices designed to securely manage, process, and store cryptographic keys. Integrating HSMs with Secret Server enhances the security of secret management by providing a secure environment for cryptographic operations and key management. This integration offers several benefits, including enhanced security, compliance with regulatory requirements, and optimized performance for cryptographic operations.

Secret Server integrates with hardware security modules (HSMs). When Secret Server is configured to use an HSM, the Secret Server encryption key is protected by that HSM.

HSMs offer several security features that traditional servers cannot. Depending on the model and design of the HSM, most HSMs are designed to be physically tamper-proof. HSMs may also be independent hardware on a network, which allows physically placing the HSM is a more secure location that might otherwise be too inconvenient for a server.

To provide broad support for HSMs, Secret Server supports any HSM that can be configured with Microsoft's Cryptography Next Generation (CNG) provider or Public-Key Cryptography Standards #11 (PKCS #11). CNG is a layer provided by Windows Server 2008 and later that HSM manufacturers can interface with. PKCS #11 is an API provided by each HSM vendor that Secret Server can interface with to perform cryptographic operations. If your HSM properly supports CNG or PKCS #11 and supports compatible algorithms, Secret Server can use it.

> Turning off HSM (deselecting the check box) in Secret Server may cause a "Server connection unavailable" error. If this happens, a manual reset of the IIS server should take care of it.

> CNG provider installation and configuration varies from HSM to HSM; however, documentation is available from each HSM vendor on how to correctly install CNG providers or set up PKCS #11.

Key Benefits of HSM Integration

1. **Enhanced Security**: HSMs significantly reduce the risk of key compromise by using a hardware-based solution for key management. They are often designed to be physically tamper-proof and can be placed in secure locations.
2. **Compliance**: HSMs meet stringent security standards and compliance requirements, helping organizations adhere to regulatory mandates.
3. **Performance**: HSMs are optimized for cryptographic operations, providing high performance and reliability.

Supported HSMs and Standards

Secret Server supports any HSM that can be configured with Microsoft's Cryptography Next Generation (CNG) provider or Public-Key Cryptography Standards #11 (PKCS #11). Some of the compatible HSMs include:

- Amazon CloudHSM
- Entrust nShield HSM
- Securosys Primus-E20
- Thales Luna Network HSM
- Utimaco CryptoServer
- Yubico YubiHSM 2

By integrating HSMs with Secret Server, organizations can achieve a higher level of security and compliance for their secret management processes.

Using Hardware Security Modules

HSM are used for Secret Server on-premise version only, for Secret Server Cloud see AWS Key Management in Secret Server Cloud.

HSM Requirements

Each HSM must provide support for these algorithms:

For CNG:

- **RSA 4096**: Support for RSA with 4096-bit keys is required. The HSM must also support RSA for encryption and decryption, in addition to signing.
- **Padding Type:** The HSM must support one of the following for RSA encryption:
  - PKCS#1 v1.5 padding
  - OAEP padding
- **CNG API Calls**:
  - NCryptCreatePersistedKey
  - NCryptDecrypt
  - NCryptDeleteKey
  - NCryptDeriveKey
  - NCryptEncrypt
  - NCryptEnumStorageProviders
  - NCryptExportKey
  - NCryptFinalizeKey

- NCryptFreeBuffer
- NCryptGetProperty
- NCryptImportKey
- NCryptIsAlgSupported
- NCryptOpenKey
- NCryptOpenStorageProvider
- NCryptSecretAgreement
- NCryptSetProperty

For PKCS #11:

AES 256: Support for AES 256-bit keys is required. The HSM must also support a CKM_AES_CBC_PAD mechanism for encryption and decryption and the slot must be a hardware slot.

Additionally, closely follow the requirements and recommendations of the HSM vendor for things such as minimum latency, redundancy, and operating environment.

> Due to limitations of the account, the NETWORK SERVICE account is not supported as an account for the IIS Application Pool. We recommend configuring Secret Server's application pool as a service account. In the advanced settings for the application pool, set "Load User Profile" to true.

> Some HSM provider's products interfere with each other. We recommend no more than one HSM provider is configured on a Windows installation at a time.

Compatible HSMs

Secret Server supports any HSM that can be configured with Microsoft's Cryptography Next Generation (CNG) provider or Public-Key Cryptography Standards #11 (PKCS #11). CNG is a layer provided by Windows Server 2008 and later that HSM manufacturers can interface with. PKCS #11 is an API provided by each HSM vendor that Secret Server can interface with to perform cryptographic operations. If your HSM properly supports CNG or PKCS #11 and supports the right algorithms, Secret Server can use it.

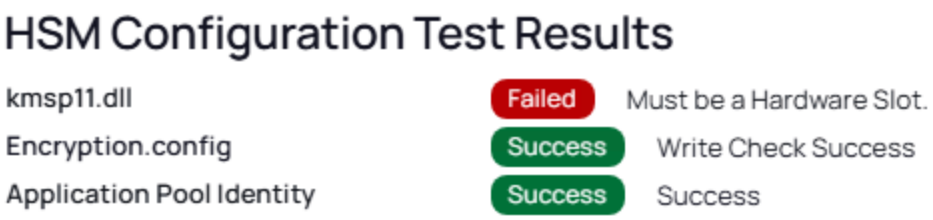
Below is a list of known HSMs that are compatible with Secret Server:

Table: Compatible HSMs

Vendor	Device	Notes
Amazon	Amazon CloudHSM (Cavium)	CNG and PKCS11 compatible with version 5.0.0 and greater. Per Amazon documentation: PKCS #11 library for AWS CloudHSM. For PKCS11 setup in Secret Server, you must enter the user pin in this format: username:password.

Vendor	Device	Notes
Entrust	nShield HSM	CNG and PKCS11 are compatible with version 12.80.4. See Entrust Integrations if you have issues with PKCS11.
Google	Google Cloud HSM	Not supported because it uses a single-key slot in the KMS for their HSM slots. Secret Server requires a hardware slot*
Securosys	Primus-E20	CNG and PKCS11 compatible with version 1.46 and greater.
Thales	Luna Network HSM	CNG with firmware 7.7.0 and greater. FIPS mode requires OAEP padding type. PKCS11 is compatible with firmware 7.4 and greater.
Thales	Luna Cloud HSM (DPoD)	CNG with firmware 7.7.0 and greater. FIPS mode requires OAEP padding type. PKCS11 is compatible with firmware 7.4 and greater.
Utimaco	CryptoServer	CNG and PKCS11 are compatible with version 4.10.0. Version 5.1.1.1 is not compatible–fails with CC mode.
Yubico	YubiHSM 2 FIPS v2.2	CNG is compatible with version 4.10.0. PKCS11 is not compatible with version 5.1.1.1–does not support AES CBC mechanism.

* Produces this error:



Silent HSM Operation

Because Secret Server is a Web application with no one physically present at the server at most times, Secret Server interacts with the HSM in "silent" mode. This prevents the HSM from attempting to interact with any users logged onto the server.

Some HSM features require interaction. If the HSM is configured in a way that requires interaction, Secret Server cannot communicate with the HSM and fails during the configuration steps.

For example, Operator Card Sets (OCS) in Thales network HSMs are such a configuration. If the Thales CNG provider is configured to use an OCS for key protection instead of module protection, someone must be physically present at the HSM and the server to insert their operator card when the key is needed. If the OCS quorum is more than a single card, Secret Server cannot interact with the HSM because it requires inserting and removing the OCS cards.

In that case, we recommend that Thales' CNG provider is configured to use module protection instead of an OCS. It is possible to use an OCS with Secret Server if the quorum is exactly one card and the card is left in the HSM at all times.

Consult your HSM vendor and their documentation to ensure that the HSM and their CNG provider are able to operate in silent mode and are configured to do so.

Configuring HSM Integration

To configure the HSM integration, go to the **Admin > HSM**.

For CNG:

You can find the list of available CNG Providers by querying for the list of registered CNG providers. Each provider must correctly report that it is a "Hardware" provider, and that it is not a Smart Card reader. If an error occurs while querying the CNG provider for its properties, it will not appear in the list; however the error is reported to Secret Server's system log. If the desired CNG provider does not appear in the list of CNG providers, ensure that the provider is correctly registered and that IIS has been restarted after the CNG registration. Also check that an error is not occurring while querying the HSM by examining the system log.

For PKCS #11:

Each HSM vendor provides a cryptoki library (dll) for PKCS #11 that Secret Server can interface with to perform cryptographic operations. Please note:

- This DLL must be copied to `c:\inetpub\wwwroot\SecretServer\pkcs11` folder for Secret Server to see it.
- This DLL must support CKM_AES_KEY_GEN and CKM_AES_CBC_PAD mechanisms and the slot must be a hardware slot.

Along with the DLL, a token label and the user PIN are needed for Secret Server setup. Please refer to your HSM vendor's documentation for obtaining these values.

Once the options are selected, Secret Server simulates encryption and decryption operations and verifies the results to check that it is functioning properly. Finally, Secret Server verifies the selected providers, and then enables HSM integration.

Stopping Other Nodes in a Clustered Environment

HSMs cannot be enabled, disabled, or rotated, in a clustered Secret Server environment. Follow these steps to stop the other nodes while updating the HSM configuration:

1. Follow these steps on all nodes except for one:
    a. Open IIS Manager on server node.
    b. Expand the server.
    c. Select **Application Pools**.
    d. Right-click on the Secret Server application pool.
    e. Select **Stop**.
2. Follow these steps on the one running node:

a. Navigate to **Admini > Backup**.
b. Click on **Run Backup Now** at the top right. This backs up the Secret Server database and application files.
c. Navigate to **Admin > HSM**.
d. Click on these HSM options: **Enable HSM**, **Disable HSM**, or **Rotate HSM Key** and follow the previous instructions.
e. Once completed, go back to IIS Manager and recycle the Secret Server application pool, or open a command prompt window as an administration and perform an IIS reset.
f. Confirm that accessing secrets in Secret Server is successful.

3. Follow these steps on each of the other server nodes:
   a. Copy the updated `encryption.config` file from the one running node to the `c:\inetpub\wwwroot\SecretServer` folder.
   b. Open IIS Manager.
   c. Expand the server.
   d. Select **Application Pools**.
   e. Right-click on the Secret Server application pool.
   f. Select **Start**.
   g. Confirm that logging in and accessing secrets in Secret Server is successful.

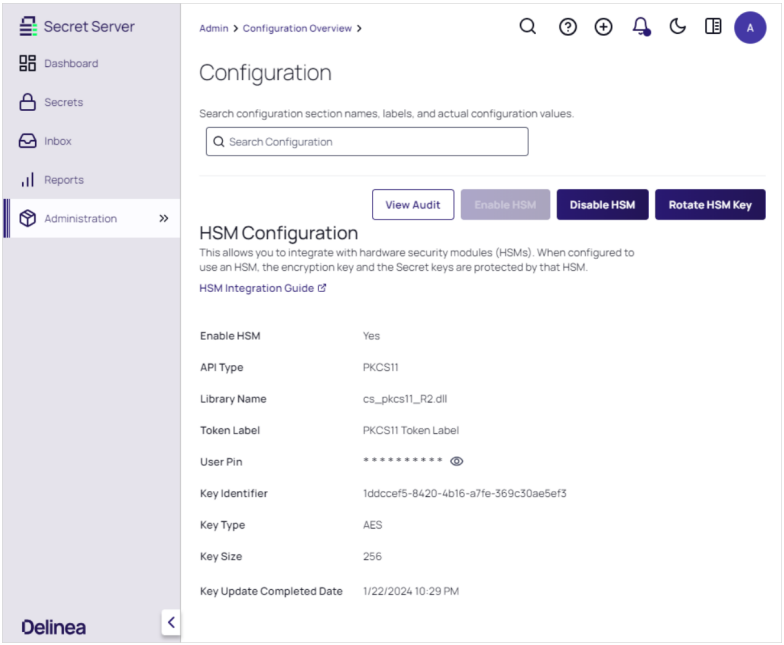
Enabling HSM

To enable the HSM:

1. Navigate to **Admin > HSM**.
2. Click **Enable HSM** at the top right.
3. **Important:** back up the `encryption.config` file before proceeding.
4. Click the **Next** button.
5. Select the **API type** - the type of API used to access HSM:
   - For CNG, select:
     - **Persistent Provider**: The location where the encryption key will be persisted.
     - **Key Type**: The type of key.
     - **Key Size**: The size of key in bits.
     - **Padding Type**: The padding scheme used for encryption with the RSA key.
   - For PKCS11, select: the **Library Name** and type the token label and user PIN.
     - **Library name**: The HSM vendor specific cryptoki library used for PKCS#11.
     - **Token Label**: The token label created during PKCS#11 set up with the HSM vendor.
     - **User pin**: The user pin created during PKCS#11 set up with the HSM vendor.

- **Key Type**: The type of key.
- **Key Size**: The size of key in bits.

6. Click **Next**. This performs a test to ensure the HSM can be used.
7. Click the **Next** button.
8. Verify the HSM configuration.
9. Click the **Save** button. The enabled configuration looks like this:



10. Do an IIS reset or application pool recycle. This starts the rotation.

Rotating the HSM Key

If HSM is enabled, Secret Server can now rotate the HSM key to ensure the secret keys are always protected by an HSM key. Rotating the HSM key only decrypts the secret keys and then re-encrypts them with the new HSM key. We recommend performing a secret key rotation after the HSM key has been rotated.

To rotate the HSM key:

1. Navigate to **Admin>HSM**.
2. Click the **Rotate HSM Key** button.

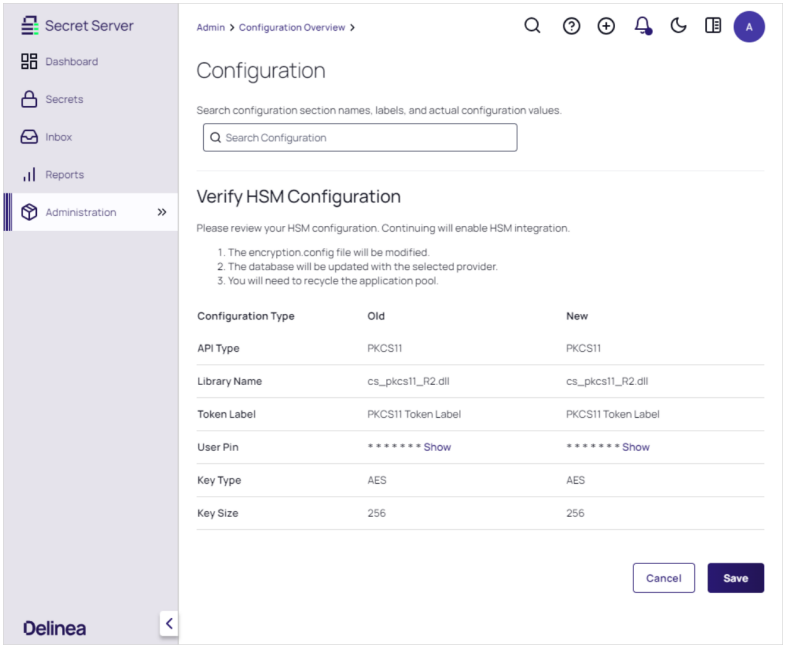
> Back up the `encryption.config` file before proceeding.

3. Click the **Next** button.
4. Select the **API type** - the type of API used to access HSM:

> Secret Server is unable to rotate to a different API type. So the selection will be the same API type as currently used.

- For CNG, select:
  - **Persistent Provider**: The location where the encryption key will be persisted.
  - **Key Type**: The type of key.
  - **Key Size**: The size of key in bits.
  - **Padding Type**: The padding scheme used for encryption with the RSA key.
- For PKCS11, select: the **Library Name** and type the token label and user PIN.
  - **Library name**: The HSM vendor specific cryptoki library used for PKCS#11.
  - **Token Label**: The token label created during PKCS#11 set up with the HSM vendor.
  - **User pin**: The user pin created during PKCS#11 set up with the HSM vendor.
  - **Key Type**: The type of key.
  - **Key Size**: The size of key in bits.

5. Click **Next**. This performs a test to ensure the HSM can be used.
6. Verify the new HSM configuration:



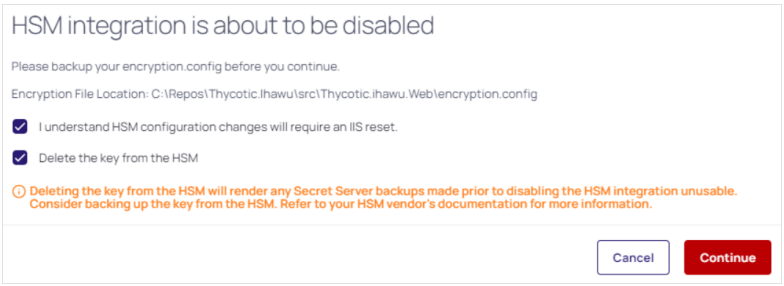
7. Click **Save**.
8. Do an IIS reset or application pool recycle. This starts the rotation.

Disabling HSMs

To disable an HSM:

This procedure requires an IIS reset.

1. Navigate to **Admin >HSM**.
2. **Important:** back up the `encryption.config` file before proceeding.
3. Click the **Disable HSM** button. A warning popup appears:



4. Decide if you want to delete the key from the HSM. Keep in mind deleting the key makes any backups using this key unusable.
5. Click **Continue**.
6. Do an IIS reset. This starts the rotation.

Securing HSM Integration

The wizard to enable, rotate, and disable HSM integration is protected by the "Administer HSM" role permission in Secret Server. The permission should be carefully assigned–if at all. Additionally, you can create an event subscription that sends alerts when the role permission is assigned or unassigned from a role.

Configuring the HSM also has its own event subscriptions for when the HSM integration is enabled, rotated, or disabled.

Additionally, an you can add an application setting to Secret Server to prevent changes to HSM configuration. Enabling, rotating, and disabling this requires direct access to the file system where Secret Server is installed.

To enable this, edit the `web-appSettings.config` file within Secret Server to contain a key called **LockHsmConfiguration** with a value of **True** as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<appSettings>
  <add key="LockHsmConfiguration" value="True" />
</appSettings>
```

This prevents access to the HSM configuration pages, regardless of role permissions. The only way to gain access is to remove this setting, thus proving you, at a minimum, have access to the server where Secret Server is installed.

HSM Redundancy

HSM redundancy varies from HSM to HSM. Please refer to the vendor's documentation on how to back up the HSM. Backups are typically either made to common file location, another HSM, or onto a smart card with the HSM's built-in smart card reader.

As long as the HSM provider is installed on the server and a key exists on the HSM with the same identifier, Secret Server attempts to use that key.

Testing HSM CNG Configuration

Secret Server does its own testing and verification of the HSM and its CNG provider before the HSM integration can be enabled. To further diagnose any issues with the HSM, the **certutil** command line utility, which is part of Windows, can test the HSM with the **-csptest** option specified. An example output may contain something like this:

```
Provider Name: SafeNet Key Storage Provider
   Name: SafeNet Key Storage Provider
……
Asymmetric Encryption Algorithms:
  RSA
      BCRYPT_ASYMMETRIC_ENCRYPTION_INTERFACE -- 3
      NCRYPT_ASYMMETRIC_ENCRYPTION_OPERATION -- 4
      NCRYPT_SIGNATURE_OPERATION -- 10 (16)
    NCryptCreatePersistedKey(SafeNet Key Storage Provider, RSA)
        Name: cngtest-6166f8fe-8caf-4e30-8e5c-a-24575
……
Pass
```

Examine the output of the test by looking for your CNG provider's name for your HSM and verifying the result. We recommend running this test using the same account as the application pool Secret Server is using. If the testing tool reports errors, consult your HSM's vendor or documentation for resolution.

PKCS #11 Log

The PKCS #11 logs are at `c:\inetpub\wwwroot\SecretServer\pkcs11\log\SSPKCS11.log`.

This captures the PKCS #11 activity between Secret Server and the HSM when enabling, rotating, and disabling.

Also, each HSM vendor will have a way to output the PKCS #11 logging to a file. Please refer to your HSM vendor's documentation on how to set up the logging.

Troubleshooting

If this error message "Thycotic.HSM.CNG.CngException: Failed to operate transformation" occurs, do the following:

1. Go to the Secret Server directory.
2. Open the `web-appSettings.config` file located at `c:\inetpub\wwwroot\SecretServer`.

3. Add the following tag in that file:

```
<?xml version="1.0" encoding="utf-8" ?>
<appSettings>
    <add key="RecycleAppPoolOnHSMError" value="true" />
</appSettings>
```

4. Restart IIS.

Enabling and Disabling HSM for Clustered Environments With Licensing Issues

> This topic only applies to **Secret Server On-Premises**.

If you are having a license issue with HSM enabled, you will need to disable HSM in order to continue to the Secret Server License page. This guide will also be helpful if you do not have a license issue, but would like to disable an HSM in Secret Server.

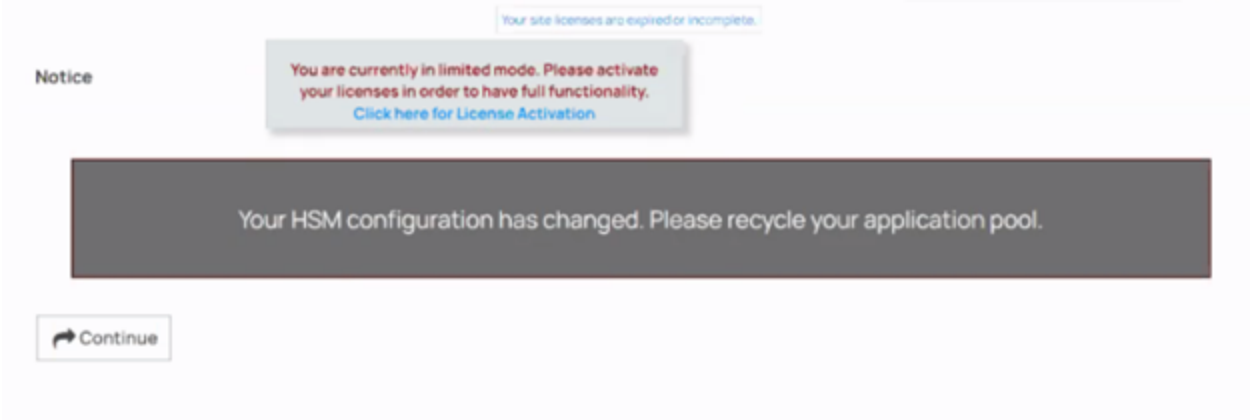
1. The first step is to backup your application folder on any of your server nodes and your database. The application folder path in *C:\Inetpub\wwroot\SecretServer*. This Secret Server folder will be the folder of the node that will not have the application pool stopped. This will be the application folder for the working node. The Secret Server folder will hold the encryption.config file. For backing up your database, you can follow the documentation from Microsoft.
2. Once the backup is completed and stored somewhere safely, please proceed and stop the Application Pools on all server nodes except for one.

   > If you have only one server node then this step can be skipped.

3. After completing step 2, please click **Disable HSM** which will give you two checkbox options: One asking you to backup the encryption.config key and one asking if you would like to clear your HSM key. Please click on Option 1 as you have completed this by backing up your Secret Server Application Folder.

   > You can choose Option 2, but this will require you to set up the HSM key again.

4. After this has completed you will get a message like in the screenshot below. This will require you to restart your application pool on the current standing node or you can run an iisreset in CMD (Run As Administrator).



5. If you are not experiencing a license issue, please skip to step 8. If you are experiencing a license issue please go to the license page.
6. At the license page, start your application pool on all application servers. If you have a single node then please disregard this step.
7. Add any new licenses to this page and make sure you click license activation to activate across all servers. Please remove any expired licenses to avoid any confusion in the future. Removing expired licenses will not effect your environment.
8. Take the backup encryption file from your working node and copy and paste it to the rest of the Application Server nodes in this file directory *C:\\Inetpub\wwroot\SecretServer*. You need to copy and replace this. Then, run an *iisreset* for each server this was replaced on.
9. Please then enable clustering if it was disabled in **Administration > Server Nodes.**
10. Please take any servers out of maintenance mode if necessary.

Master Encryption Key Rotation

Important Support Note: Rotating the master encryption key is a sensitive operation. While we have detailed the steps in this topic, we urge you to contact Support to work with a support representative before attempting any MEK rotation to ensure all prerequisites have been met. Please provide at least a five-day notice in advance of when you want to conduct the rotation.

Overview

When Secret Server is first installed, a unique random AES256 Master Encryption Key (MEK) is generated and saved in a file, `encryption.config`. The MEK protects anything sensitive in Secret Server that is not associated with a specific secret, as well as each secret's unique AES256 key when an HSM is not used.

For added security, you can rotate the MEK, re-encrypting protected data with the new key.

Geo-Replication, DPAPI, or an active secret key rotation must not be in effect when rotating the MEK.

MEK rotation fully supports using a Hardware Security Module (HSM) for Secret Server On-Premises or key management for Secret Server Cloud. This feature requires Enterprise Plus edition or the advanced encryption license. If you are using an HSM, disable it for your first MEK rotation. You can then re-enabled it, and you can run future MEK rotations without disabling it.

If the rotation button is disabled then DPAPI also needs to be disabled, otherwise it will cause the process to fail. The DPAPI needs to be disabled on all nodes.

Rotation Procedure

To perform a MEK rotation:

1. **Important:** For Secret Server on Premises, back up the current MEK file `encryption.config` in the application directory **on each node**.

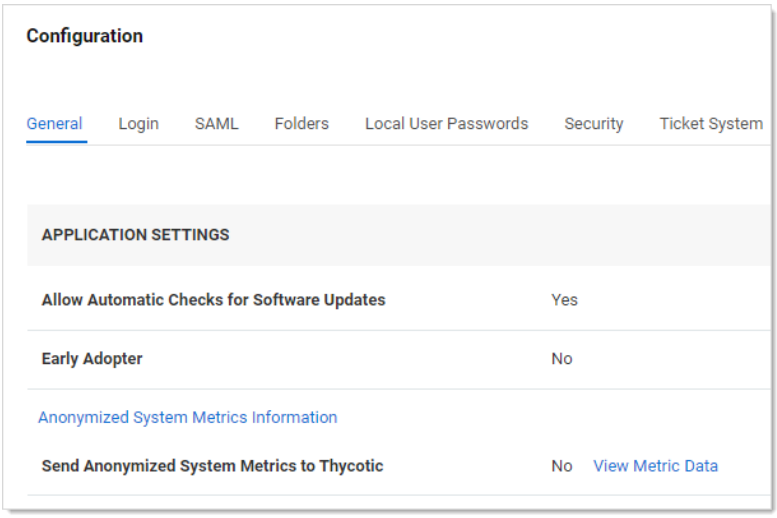
   Do not continue until you back up your encryption key. Failure to do so could cause extensive permanent data loss.

   For Secret Server on Premises, if this is the first MEK rotation, also backup any encrypted session recording videos saved to disk. They are only updated during the very first MEK rotation. Failure to do so could cause extensive **permanent data loss**.

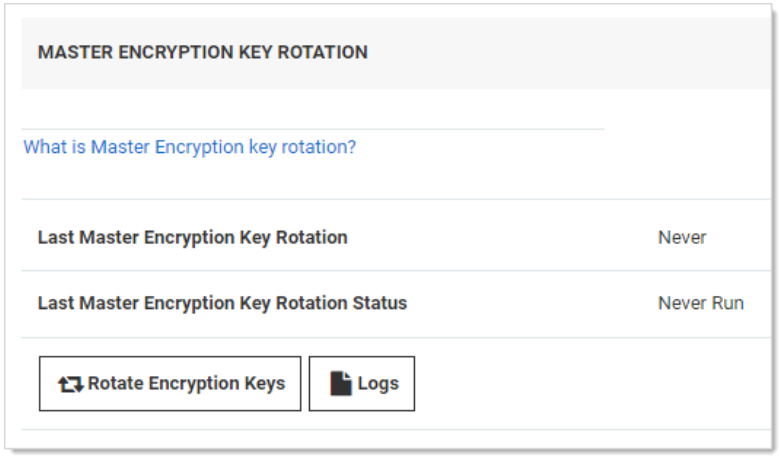
2. **Important:** For Secret Server on Premises, back up your database. See the first task on the "Moving the Microsoft SQL Server Database to Another Machine" on page 101 topic for details.

   Do not continue until you back up your database. Failure to do so could cause extensive **permanent data loss**.

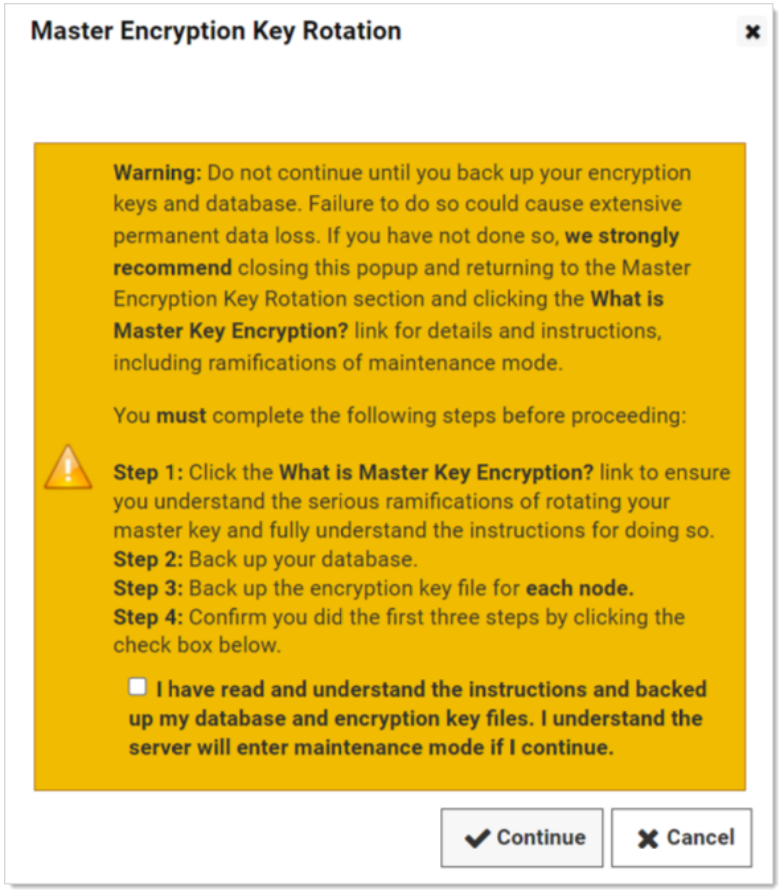
3. Go to **Admin > Configuration > Security**. The Configuration page appears:



4. Click the **Security** tab.
5. Go to the **Master Encryption Key Rotation** section (*not* the Key Rotation section):



6. Click the **Rotate Encryption Keys** button. The Master Encryption Key Rotation popup appears:



7. Carefully read the text, especially for the check box itself.

8. When you are finished, click to select the check box to acknowledge having read it.
9. Click the **Continue** button. The server goes into maintenance mode to perform MEK rotation. Secrets and configuration settings cannot be updated while in maintenance mode. Secrets cannot be updated while in this mode. Processing time will vary, depending on the hardware, number of secrets, and HSM key size.
10. The rotation begins.
11. Secret Server updates the `encryption.config` file on the current server. The file now contains the new MEK, as well as the previous MEK, which is needed to re-encrypt all of the data. The `encryption.config` file is only updated on the node you are connected to when you click the **Rotate Encryption Keys** button, so, depending on whether you have clustered servers, pick one of the following procedures:

    If you are *not* running a cluster of On-Premises servers:

    a. When prompted to restart the IIS application, run `iisreset` to stop and restart the IIS server.

    b. Open Secret Server to restart it.

    If you *are* running a cluster of On-Premises servers:

    a. When prompted to restart the IIS server, run command `iisreset /stop` on all nodes in an elevated command prompt. This stops but does not restart IIS on the nodes.

    b. Copy the `encryption.config` file from the updated node to all other nodes.

    Note: If you are not sure which node performed the rotation, check the "modified time" of the `encryption.config` files.

    c. Run `iisreset /start` on all nodes. This restarts IIS on the nodes.

    d. Open Secret Server to restart it. The first Secret Server node you run continues the rotation process.
12. The new MEK takes effect, and any *new* records encrypted use the new key. Any data still using the old key continues to work until the rotation fully finishes.
13. Secret Server's background worker (Secret Server-BWSR.log) enables maintenance mode, preventing changes to secrets and other configuration settings, and continues the rotation process.
14. Secret Server marks the data needing an update in the database and begins rotating everything. A progress bar appears on the **Security** tab.
15. After all data is rotated except for session recording videos, maintenance mode is disabled.
16. Secret Server's session recording role (Secret Server-SRWSR.log) starts rotating any existing session recording videos encrypted by the MEK. The **Last Master Encryption Key Rotation Status** text changes to **Pending - Encrypting Session Data**. Depending on the number of videos, this may take some time. This only applies to the first MEK rotation–further rotations do not update the recordings again.
17. Once the process finishes, the **Last Master Encryption Key Rotation Status** text changes to **Completed - Master Encryption Keys Rotated**.

> Because we only track the current and previous MEK, the MEK rotation must be fully successful before you can run another one. A Retry button appears if the process times out or needs restarting. Monitor the rotation status and contact support if it fails to complete. Failure to do so could cause extensive **permanent data loss**.

Secret Key Rotation

Overview

Secret key rotation is a somewhat similar process to RPC by which the encryption key, used for securing secret data, is changed and that secret data is re-encrypted. Each secret receives a new, unique AES-256 key. Secret key rotation can be used to meet compliance requirements that mandate encryption keys be changed on a regular basis.

How to Perform Secret Key Rotation

Secret key rotation requires the Rotate Encryption Keys permission.

1. Go to **Admin > Configuration > Security.**
2. In the **Key Rotation** section, click the **Rotate Secret Keys** button.

Secret key rotation begins as soon as Secret Server enters maintenance mode. Because maintenance mode disables various functionality (such as secrets cannot be updated), the timing of secret key rotation merits consideration of Secret Server processing time. We recommend running secret key rotation during off-peak or non-business hours.

To learn more about maintenance mode, see the "Maintenance Mode FAQ" on page 217.

Estimated Processing Time

Maintenance mode takes five minutes to enable before secret key rotation is started. The processing time for secret key rotation varies greatly, depending on the following factors:

- Total number of secrets
- Total number of secrets with file attachments and the size of those file attachments
- Hardware configuration:
  - Number of CPUs and cores
  - Memory size
  - Network latency
- HSM key size, if applicable

As a general guideline, use the following:

Table: Secret Key Rotation Processing Time

Configuration	Approximate Time Taken
Without HSM (default)	2,000-12,000 secrets per minute

Configuration	Approximate Time Taken
HSM with a 2048-bit key	240-600 secrets per minute
HSM with a 4096-bit key	120-300 Secrets per minute

Secret Server Telemetry

This topic only applies to Secret Server On-Premises

Overview

There are 3 reasons for Delinea products to call home–when:

- Checking for available updates
- Activating licenses
- Reporting anonymized usage metrics

Each of these communications is explained below and can be disabled or avoided.

Checking for and Downloading Updates

Frequency: Once per day

The software checks for available updates and sends the following information to Delinea's update server:

- .NET Framework version
- IP address of the installed instance
- Microsoft SQL Server version
- Microsoft Windows version
- Product version

Checking for updates and sending this information will only occur if both of the following are true:

- The server has outbound network access, which you can block at a firewall.
- The "Allow Automatic Checks for Software Updates" check box is enabled at **Admin > Configuration > Application** (see below).

No sensitive data is sent during the check. Its only purpose is to alert administrators if a software update is available. The queried website is also used to download new software versions during the upgrade process. If you wish to allowlist the specific servers involved, they are:

- `d36zgw9sidnotm.cloudfront.net:443`
- `updates.thycotic.net:443`
- `updates.thycotic.net:80`
- `tmsnuget.thycotic.com/nuget/`

Note that Secret Server would still manually check for an update if there is internet access and a manual update is triggered, even if Allow Automatic Checks for Software Updates feature is off.

License Activation

Frequency: when a new license is activated.

The software also sends contact and license-key information, provided by the administrator, to Delinea during online license activation. The same information is sent via another computer for offline activation.

Reporting Anonymized Usage Metrics

This section only applies to Secret Server and Secret Server Cloud versions 10.6 and above.

Delinea collects anonymized usage data to help guide future research and development plans so that product improvements can provide the greatest benefit to customers.

Frequency: Once per day

Secret Server returns anonymized metrics across several categories:

- A unique identifier number that allows Delinea to correlate metrics from the same server over time but does not contain any information that identifies the customer.
- License information, including edition information and the number of licensed users but not license keys or other identifying data.
- Product configuration and usage, such as number of secrets stored and product feature status, not including any identifying data.
- Product environment, including host operating system and SQL server version, not including any identifying data.

Reporting of anonymized metrics only occurs if:

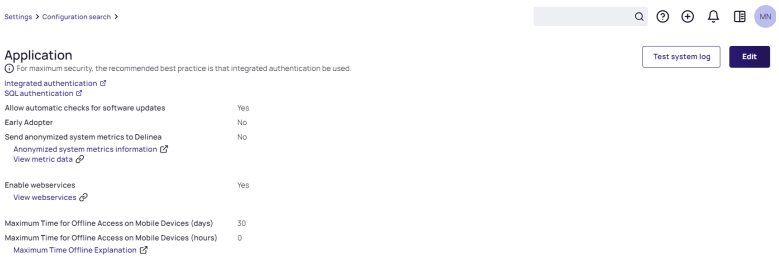
- The server has outbound network access (you can block your server at a firewall if desired)
- The "Send Anonymized System Metrics to Delinea" setting under Admin > Configuration is enabled (see below).

You can allow for the metrics reporting on your firewall by allowlisting: `https://telemetry.thycotic.net:443`.

Setting and Viewing Secret Server Telemetry

To set or view telemetry:

1. Click **Admin > Configuration search**, then select **Application** from the General section. The Applicationsettings page appears:



2. (Optional) To view the JSON file for the possible sent metrics, click the **View Metric Data** link. The file appears:


```
{
  "identifier": "cef588896e3e9718d59ccdd0f9e77568",
  "licenses": [],
  "features": [
    {
      "name": "Remote Password Changing",
      "enabled": true,
      "count": 0,
      "countDescription": "Secrets with AutoChange enabled"
    },
    {
      "name": "Heartbeat",
      "enabled": true,
      "count": 614,
      "countDescription": "Secrets with Heartbeat enabled"
    },
    {
      "name": "Checkout",
      "enabled": true,
      "count": 5,
      "countDescription": "Secrets with Checkout enabled"
    },
    {
      "name": "Checkout Change Password",
      "enabled": false,
      "count": 0,
      "countDescription": "Secrets with Checkout change password enabled"
    },
    {
      "name": "DoubleLock",
      "enabled": true,
      "count": 0,
      "countDescription": "Secrets with DoubleLock enabled"
    },
    {
      "name": "Request Access",
      "enabled": true,
      "count": 104,
      "countDescription": "Secrets with Request Access enabled"
    },
    {
      "name": "Request Access Editors need approval",
      "enabled": true,
```


3. Click **Edit**. The section changes to edit mode.
4. Click to select or deselect the **Send Anonymized System Metrics to Delinea** check box.
5. Scroll down and click the **Save** button.

Securing ASP Cookies

This topic only applies to **Secret Server On-Premises**.

To secure your ASP session and forms authentication cookies, perform the following steps:

1. Ensure that there is an SSL certificate installed for the instance.
2. Log in to Secret Server using HTTPS.
3. Navigate to the **Admin > Configuration** page
4. Click on the **Security** tab.
5. Click the **Edit** button
6. Check the **Force HTTPS/SSL** check box
7. Click the **Save** button.
8. Open the `web-cookie.config` file in the application installation folder.
9. Set `requireSSL` to `true`.

   Save and Close the file.
10. Open the `web-auth.config`file in the application installation folder.
11. Set `requireSSL` to `true` . If the attribute does not exist, add it to the `forms` tag.

    Save and Close the file.
12. Recycle the Secret Server's application pool.

> If you later migrate Secret Server to a new server, SSL must be configured on the new server before you can log in due to these settings. If you want to log in prior to configuring SSL, reverse steps 8 through 13 and recycle the application pool.

Securing IIS Server

> This topic only applies to **Secret Server On-Premises**.

This is a list of items that IIS admin can implement to secure the IIS Web server for additional Secret Server hardening.

Accounts

- Remove unused accounts from the server.
- Disable the Windows Guest account.
- Rename the Administrator account.
- Ensure the Administrator account has a strong password.
- Ensure the IUSR_MACHINE account is disabled if it is not used by the application.
- If your applications require anonymous access, create a custom least-privileged anonymous account. Ensure the anonymous account does not have write access to Web content directories and cannot execute command-line tools.
- Ensure the ASP.NET process account is configured for least privilege. This only applies if you are not using the default ASPNET account, which is a least-privileged account.

- Ensure strong account and password policies are enforced for the server.
- Restrict remote logons–the "Access this computer from the network" user-right is removed from the Everyone group.
- Disable null sessions (anonymous logons).
- Ensure no more than two accounts are in the Administrators group.

Auditing and Logging

- Audit failed logon attempts.
- Relocate and secure IIS log files.
- Configure log files with an appropriate size, depending on the application security requirement.
- Regularly archive and analyze log files.
- Audit access to the `Metabase.bin` file.
- Configure IIS to us the W3C extended log file format for auditing.

Code Access Security

- Enable code access security on the server.
- Remove all permissions from the local intranet zone.
- Remove all permissions from the Internet zone.

Files and Directories

- Ensure files and directories are contained on NTFS volumes.
- Ensure Web site content is located on a non-system NTFS volume.
- Ensure log files are located on a non-system NTFS volume and not on the same volume where the Web site content resides.
- Ensure the All Vault Users group is restricted (no access to `\Windows\system32` or Web directories).
- Ensure the website root directory has deny write ACE for anonymous Internet accounts.
- Ensure content directories have deny write ACE for anonymous Internet accounts.
- Remove the Remote IIS administration application.
- Remove the Resource Kit tools, utilities, and SDKs.

IIS Metabase

- Use NTFS permissions to restrict access to the metabase (`%systemroot%\system32\inetsrv\metabase.bin`).
- Ensure IIS banner information is restricted (IP address in content location is disabled).

ISAPI Filters

Ensure unnecessary or unused ISAPI filters are removed from the server.

Machine.config

- Ensure protected resources are mapped to HttpForbiddenHandler.
- Remove unused HttpModules.
- Ensure tracing is disabled: `<trace enable="false"/>`.
- Ensure debug compiles are turned off: `<compilation debug="false" explicit="true" defaultLanguage="vb">`

Patches and Updates

- Run Microsoft Baseline Security Analyzer on a regular interval to check for latest operating system and components updates, including

  Windows, IIS server, and the .NET Framework.
- Test updates on development servers prior to deployment on production servers.
- Check the Microsoft Security Notification Service at docs.microsoft.com on a regular interval for up-to-date Microsoft technical security notifications.

Ports

- Ensure Internet-facing interfaces are restricted to port 80 (and 443 if SSL is used).
- Ensure Intranet traffic is encrypted (for example, with SSL) or restricted.

Protocols

- Disable WebDAV if not used by the application or secure it if it is required.
- Harden the TCP/IP stack.
- Ensure NetBIOS and SMB are disabled if not used (closes ports 137, 138, 139, and 445).

Registry

- Restrict remote registry access.
- Secure SAM (`HKLM\System\CurrentControlSet\Control\LSA\NoLMHash`).

Script Mappings

- Ensure extensions not used by the application are mapped to `404.dll`, including .idq, .htw, .ida, .shtml, .shtm, .stm, idc, .htr, and .printer.
- Ensure unnecessary ASP.NET file type extensions are mapped to HttpForbiddenHandler in `Machine.config`.

Server Certificates

- Ensure certificate date ranges are valid.
- Ensure certificates are used for their intended purpose (for example, the server certificate is not used for e-mail).

- Ensure the certificate's public key is valid, all the way to a trusted root authority.
- Ensure the certificate is SHA 256 or better.

Services

- Disable unnecessary Windows services.
- Ensure services are running with least-privileged accounts.
- You can disable FTP, SMTP, and NNTP services if they are not required.
- Ensure the Telnet service is disabled.
- Ensure the ASP.NET state service is disabled and is not used by your applications.

Shares

- Ensure all unnecessary shares are removed (including default administration shares).
- Restrict access to required shares (the All Vault Users group does not have access).
- Remove administrative shares (C$ and Admin$) if they are not required.

Sites and Virtual Directories

- Ensure Web sites are located on a non-system partition.
- Ensure the "Parent paths" setting is disabled.
- Remove potentially dangerous virtual directories, including IISSamples, IISAdmin, IISHelp, and Scripts.
- Remove or secure MSADC virtual directory (RDS).
- Ensure include directories do not have the "Read Web" permission.
- Restrict Write and Execute Web permissions for the anonymous account on virtual directories that allow anonymous access.
- Ensure there is script source access only on folders that support content authoring.
- Ensure there is write access only on folders that support content authoring and these folder are configured for authentication (and SSL encryption, if required).
- Remove FrontPage Server Extensions (FPSE) if not used. If they are used, ensure they are updated and access to FPSE is restricted.

Other Considerations

- Ensure server remote administration is secured and configured for encryption, low session time-outs, and account lockouts.Ensure HTTP requests are filtered.
- Use a dedicated machine as a Web server.
- Physically protect the Web server machine in a secure machine room.
- Configure a separate anonymous user account for each application, if you host multiple Web applications.
- Do not install the IIS server on a domain controller.

- Do not connect an IIS Server to the Internet until it is fully hardened.
- Do not allow anyone except the administrator to locally log on to the machine.

Securing IIS Server

> This topic only applies to **Secret Server On-Premises**.

This is a list of items that IIS admin can implement to secure the IIS Web server for additional Secret Server hardening.

Accounts

- Remove unused accounts from the server.
- Disable the Windows Guest account.
- Rename the Administrator account.
- Ensure the Administrator account has a strong password.
- Ensure the IUSR_MACHINE account is disabled if it is not used by the application.
- If your applications require anonymous access, create a custom least-privileged anonymous account. Ensure the anonymous account does not have write access to Web content directories and cannot execute command-line tools.
- Ensure the ASP.NET process account is configured for least privilege. This only applies if you are not using the default ASPNET account, which is a least-privileged account.
- Ensure strong account and password policies are enforced for the server.
- Restrict remote logons–the "Access this computer from the network" user-right is removed from the Everyone group.
- Disable null sessions (anonymous logons).
- Ensure no more than two accounts are in the Administrators group.

Auditing and Logging

- Audit failed logon attempts.
- Relocate and secure IIS log files.
- Configure log files with an appropriate size, depending on the application security requirement.
- Regularly archive and analyze log files.
- Audit access to the `Metabase.bin` file.
- Configure IIS to us the W3C extended log file format for auditing.

Code Access Security

- Enable code access security on the server.
- Remove all permissions from the local intranet zone.

- Remove all permissions from the Internet zone.

Files and Directories

- Ensure files and directories are contained on NTFS volumes.
- Ensure Web site content is located on a non-system NTFS volume.
- Ensure log files are located on a non-system NTFS volume and not on the same volume where the Web site content resides.
- Ensure the All Vault Users group is restricted (no access to `\Windows\system32` or Web directories).
- Ensure the website root directory has deny write ACE for anonymous Internet accounts.
- Ensure content directories have deny write ACE for anonymous Internet accounts.
- Remove the Remote IIS administration application.
- Remove the Resource Kit tools, utilities, and SDKs.

IIS Metabase

- Use NTFS permissions to restrict access to the metabase (`%systemroot%\system32\inetsrv\metabase.bin`).
- Ensure IIS banner information is restricted (IP address in content location is disabled).

ISAPI Filters

Ensure unnecessary or unused ISAPI filters are removed from the server.

Machine.config

- Ensure protected resources are mapped to HttpForbiddenHandler.
- Remove unused HttpModules.
- Ensure tracing is disabled: `<trace enable="false"/>`.
- Ensure debug compiles are turned off: `<compilation debug="false" explicit="true" defaultLanguage="vb">`

Patches and Updates

- Run Microsoft Baseline Security Analyzer on a regular interval to check for latest operating system and components updates, including

  Windows, IIS server, and the .NET Framework.
- Test updates on development servers prior to deployment on production servers.
- Check the Microsoft Security Notification Service at docs.microsoft.com on a regular interval for up-to-date Microsoft technical security notifications.

Ports

- Ensure Internet-facing interfaces are restricted to port 80 (and 443 if SSL is used).
- Ensure Intranet traffic is encrypted (for example, with SSL) or restricted.

Protocols

- Disable WebDAV if not used by the application or secure it if it is required.
- Harden the TCP/IP stack.
- Ensure NetBIOS and SMB are disabled if not used (closes ports 137, 138, 139, and 445).

Registry

- Restrict remote registry access.
- Secure SAM (`HKLM\System\CurrentControlSet\Control\LSA\NoLMHash`).

Script Mappings

- Ensure extensions not used by the application are mapped to `404.dll`, including .idq, .htw, .ida, .shtml, .shtm, .stm, idc, .htr, and .printer.
- Ensure unnecessary ASP.NET file type extensions are mapped to HttpForbiddenHandler in `Machine.config`.

Server Certificates

- Ensure certificate date ranges are valid.
- Ensure certificates are used for their intended purpose (for example, the server certificate is not used for e-mail).
- Ensure the certificate's public key is valid, all the way to a trusted root authority.
- Ensure the certificate is SHA 256 or better.

Services

- Disable unnecessary Windows services.
- Ensure services are running with least-privileged accounts.
- You can disable FTP, SMTP, and NNTP services if they are not required.
- Ensure the Telnet service is disabled.
- Ensure the ASP.NET state service is disabled and is not used by your applications.

Shares

- Ensure all unnecessary shares are removed (including default administration shares).
- Restrict access to required shares (the All Vault Users group does not have access).
- Remove administrative shares (C$ and Admin$) if they are not required.

Sites and Virtual Directories

- Ensure Web sites are located on a non-system partition.
- Ensure the "Parent paths" setting is disabled.
- Remove potentially dangerous virtual directories, including IISSamples, IISAdmin, IISHelp, and Scripts.
- Remove or secure MSADC virtual directory (RDS).
- Ensure include directories do not have the "Read Web" permission.
- Restrict Write and Execute Web permissions for the anonymous account on virtual directories that allow anonymous access.
- Ensure there is script source access only on folders that support content authoring.
- Ensure there is write access only on folders that support content authoring and these folder are configured for authentication (and SSL encryption, if required).
- Remove FrontPage Server Extensions (FPSE) if not used. If they are used, ensure they are updated and access to FPSE is restricted.

Other Considerations

- Ensure server remote administration is secured and configured for encryption, low session time-outs, and account lockouts.Ensure HTTP requests are filtered.
- Use a dedicated machine as a Web server.
- Physically protect the Web server machine in a secure machine room.
- Configure a separate anonymous user account for each application, if you host multiple Web applications.
- Do not install the IIS server on a domain controller.
- Do not connect an IIS Server to the Internet until it is fully hardened.
- Do not allow anyone except the administrator to locally log on to the machine.

Hiding HTTP Header Information

This topic only applies to **Secret Server On-Premises**.

Web applications, such as Secret Server, can leak information useful to attackers via headers, error messages, version numbers, and more. To hide HTTP header information in Secret Server, follow the procedures below.

Hide the IIS Version

To hide the version of IIS used on the server, remove the HTTP header *X-Powered-By* by following the steps below:

1. Open the IIS Manager.
2. In the Connections tree, select the website that Secret Server is running under.
3. Click the **HTTP Response Headers** button on the right. The HTTP Response Headers panel appears.

4. Click to select the **X-Powered-By HTTP** header.
5. Click the **Remove** button in the **Actions** panel.

Hide the ASP.NET Version

To hide the version of ASP.NET used by the Secret Server application pool, remove the HTTP header *X-ASPNET-VERSION* by following the steps below:

1. Open the `web.config` file for Secret Server, which is located in the root directory for the website.
2. Inside the `<system.web>` tag, add the tag `<httpRuntime enableVersionHeader="false"/>`.
3. Save the file.

Hide the Server Type

To hide the server type, remove the line, `Server: Microsoft-HTTPAPI/2.0` (added by the .NET framework) from the HTTP header using the procedure below:

> Although there are other methods for hiding the server type, we strongly recommend updating the Windows registry using the procedure below. Do not simply remove the server header variable. Doing so will cause parts of Secret Server to malfunction.

1. Navigate to **Computer > HKEY_LOCAL_MACHINE > SYSTEM > CurrentControlSet > Services > HTTP > Parameters**.
2. Change the **DisableServerHeader (REG_DWORD type)** registry key from 0 to 1.

APIs and Scripting

> This section contains general material on API access and scripting. APIs, scripts and techniques may appear elsewhere in the documentation for specific aspects of Secret Server. For example, Automatic Secret Export REST API is in the Secret Import and Export sections.

Secret Server offers a robust set of web services and APIs that enable organizations to seamlessly integrate privileged account management into their IT infrastructure and automate critical security processes. These APIs are designed to provide a secure and flexible way to interact with Secret Server programmatically, allowing for the development of custom workflows, integration with third-party applications, and enhancement of DevOps practices.

Web Services API

Secret Server's web services API includes both SOAP and RESTful interfaces, making it accessible to a wide range of programming languages and platforms. Developers can use these APIs to automate tasks such as creating, retrieving, and managing secrets, as well as configuring folders and permissions. This level of integration is crucial for maintaining a secure and efficient environment where privileged credentials are centrally managed and protected.

Key features of the Web Services API include:

- Authenticating to Secret Server
- Creating and managing folders
- Saving new secrets and editing existing ones
- Searching for secrets
- Setting permissions on secrets and folders

Learn more about API web services:

- API Web Services for Secret Server | Supports REST and SOAP

REST API Examples

Delinea provides examples and guides for using the REST API with various programming languages, such as Python and Perl. These examples demonstrate how to authenticate to Secret Server, retrieve secrets, and perform updates on secret fields and items. The REST API is particularly user-friendly, leveraging standard HTTP methods and status codes for operations, making it a preferred choice for modern web applications.

REST API capabilities include:

- Integration with scripting languages for custom automation
- Operations to manage secrets and their attributes
- Token-based authentication for secure access

REST API documentation:

- "REST API Reference Download" on page 1500

By using Secret Server's web services and APIs, organizations can ensure that their privileged account management is not only secure but also seamlessly integrated with their existing tools and workflows, enabling them to maintain a strong security posture while improving operational efficiency.

API Authentication

Script authentication in Secret Server is a critical aspect of securely automating tasks and integrating with other systems. The recommended approach is to use the Secret Server Software Development Kit (SDK) for DevOps, which provides a secure way to handle credentials and access tokens. The SDK ensures that credentials are stored securely and that tokens are used with a least-privilege approach, limiting access to only what is necessary. Additionally, it is important to avoid using admin accounts for scripts, never store passwords or tokens in plain text, and always set tokens to expire after a short period. Integrated Windows Authentication (IWA) can also be used for on-premises deployments. Following these best practices helps mitigate risks associated with automated scripts and ensures robust security.

Generating Self-Signed Certificates for Scripts

Please run the following as **Administrator**.

```
# This simply generates a self-signed certificate which will import into <Secret Server
URL />
# Requires .NET 4.5 or above
# Please Run As Administrator
### User Variables ###
# Filename of PFX
$filename = 'PFXNAMEHERE.PFX'
# Certificate Password for PFX
$pass = Read-Host -Prompt "Please Enter Password for .pfx file" -AsSecureString
# DNS name in certificate
$dnsname = Read-Host -Prompt "Please enter the server's FQDN"
###--Commands--###
# NOTE: The provider must be set in order to be compatible with .NET 4.5 newer versions of
.NET can import certs from more providers
try {
    $cert = New-SelfSignedCertificate -CertStoreLocation cert:\localmachine\my -DnsName
$dnsname -HashAlgorithm SHA256 -KeyLength 4096 -Provider "Microsoft Enhanced RSA and AES
Cryptographic Provider"
    $path = 'cert:\localmachine\my\' + $cert.thumbprint
    Export-PfxCertificate -Cert $path -FilePath $filename -Password $pass
    # remove from cert store
    Remove-Item $path
}
catch { Write-Error $_ }
```

Script Authentication Using Tokens

Overview

Manipulating the Secret Server API via scripts is very handy and powerful but also potentially dangerous, so it is critical to follow best practices to ensure robust security. This topic discusses these best practices. We recommend the Secret Server Software Development Kit for DevOps for all automated or machine-to-machine scripts. If not using the SDK, we strongly recommend creating an application account for script access and limiting permissions via both roles and the secrets themselves.

Best Practices

For security, it is critical that you:

- Do not use admin accounts.
- Use a least-privilege account, limiting role and secret access permissions.
- Never store passwords or tokens in plain text. Use secure credentials instead–see below for details.
- Use a least-privilege application account. These are restricted from logging in the UI.
- Never use domain admin credentials in an automated script.
- Create a specific service account with only the permissions needed to get the task done.
- Always expire tokens. To expire tokens, navigate to the **Admin > Application**, click **Edit**, check to enable **Session Timeout for Webservices**, and set the timeout to a short period (the default is 20 minutes).

Authentication Methods

> Entra ID authentication is not supported

Scripts Run by Automated Processes

Recommended methods:

- Secret Server Software Development Kit for DevOps (SDK). The SDK is a great way to limit access to a specific client with encrypted credentials and IP restrictions. See Using the Secret Server SDK for DevOps for more information.
- Integrated Windows Authentication (IWA).

Specific User Use Cases with Script Augmentation

Recommended methods:

- Get credentials with two-factor authentication (2FA). This is a secure way to prompt users for credentials for manually run scripts.
- Prompt with 2FA. This prompts the user for script credentials.
- IWA.

Example Credential-Access Scripts

Recommended Methods

Get-Credential with 2FA

```
$application = "<Secret Server URL>"
function Get-Token
{
    [CmdletBinding()]
    param(
        $credentials
        [Switch] $UseTwoFactor
    )
    $creds = @{
        username = $credentials.UserName
        password = $credentials.GetNetworkCredential().Password
        grant_type = "password"
    };
    $headers = $null
    If ($UseTwoFactor) {
        $headers = @{
            "OTP" = (Read-Host -Prompt "Enter your OTP for 2FA: ")
        }
    }
    try
    {
```

```
        $response = Invoke-RestMethod "$application/oauth2/token" -Method Post -Body
$creds -Headers $headers;
        $token = $response.access_token;
        return $token;
    }
    catch
    {
        $result = $_.Exception.Response.GetResponseStream();
        $reader = New-Object System.IO.StreamReader($result);
        $reader.BaseStream.Position = 0;
        $reader.DiscardBufferedData();
        $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
        Write-Host "ERROR: $($responseBody.error)"
        return;
    }
}
$token = Get-Token -credentials (Get-Credential) -UseTwoFactor
```

Prompting for Credentials with 2FA

```
$application = "<Secret Server URL>"
function Get-Token
{
    [CmdletBinding()]
    param( [Switch] $UseTwoFactor
    )
    $creds = @{
        username = Read-Host -Prompt "Enter your username: "
        password = Read-Host -Prompt "Enter your password: "
        grant_type = "password"
    };
    $headers = $null
    If ($UseTwoFactor) {
    $headers = @{
        "OTP" = (Read-Host -Prompt "Enter your OTP for 2FA: ")
       }
    }
    try
    {
        $response = Invoke-RestMethod "$application/oauth2/token" -Method Post -Body
$creds -Headers $headers;
        $token = $response.access_token;
        return $token;
    }
    catch
    {
        $result = $_.Exception.Response.GetResponseStream();
        $reader = New-Object System.IO.StreamReader($result);
        $reader.BaseStream.Position = 0;
        $reader.DiscardBufferedData();
        $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
```

```
        Write-Host "ERROR: $($responseBody.error)"
        return;
    }
}
$token = Get-Token -UseTwoFactor
```

Integrated Windows Authentication

> Secret Server Cloud does not support IWA.

```
$api = "<Secret Server URL>" /winauthwebservices/api/v1"
$endpoint = "$api/secrets/3844"
$secret = Invoke-RestMethod $endpoint -UseDefaultCredentials
```

> See "Using Webservices with IWA via PowerShell" on page 393 for more information.

Downloading Example Scripts

Example Scripts:

- "REST API PowerShell Scripts" on page 1503

> For REST API Client Generation (Advanced), please see "REST API Client Generation with Swagger" on page 1495

> See "Using Webservices with IWA via PowerShell" on page 393 for more information.

Secret-based Credentials for PowerShell Scripts

Overview

You can specify a secret to provide the default credentials for running all PowerShell scripts on a site. This allows sites in different data centers to have different default credentials. This applies to remote password changing, checkout hooks, and account discovery PowerShell scripts.

> If you want a specific secret checkout hook, secret password changer, or account discovery scanner to use different credentials you can still provide credentials in those areas, which will take precedence over the one set on the site.

RunAs Secret Precedence

Remote Password Changing

The precedence order for which RunAs secret to use for remote password changing is:

1. Privileged account on the secret RPC tab
2. Secret site's RunAs secret
3. Secret

Secret Dependencies

The precedence order for which RunAs secret to use for PowerShell Secret dependencies is:

1. Privileged account on the dependency
2. RunAs secret on the dependency group's site
3. Secret site's RunAs secret
4. Secret

Checkout Hooks

The precedence order for which RunAs secret to use for checkout hooks is:

1. Privileged account on the hook
2. Secret site's RunAs secret
3. Secret

Procedures

Setting the Default PowerShell Credential for a Site

To set a default PowerShell credential for a site:

1. Go to **Admin > Distributed Engine** and select a site from the list.
2. In the **Advanced site configuration** section click **Edit**.
3. Next to the **Default PowerShell RunAs Secret** field click on the **No secret selected** link and select the related secret from the list.
4. Click **Save**.

Using the Site PowerShell Credentials for Discovery

To use the site PowerShell credentials on a discovery scanner:

1. Add a PowerShell scanner to a discovery source or edit an existing scanner:
   - Navigate to **Admin > Discovery** and select the **Configuration** tab.
   - Select **Extensible Discovery** from the dropdown at the top right, and select **Configure Discovery Scanners**.
   - Select a scanner from the list and click **Edit** to edit an existing scanner, or click **Create scanner** at the top right if you would like to create a new scanner.

- If creating a new scanner, specify its **Name** and **Description**, check to enable **State**, select the related **Scanner type**, and in the **Base scanner** field select **PowerShell Discovery**.
- Optionally check to enable **Allow OU** input, select the **Input template, Output template**, and **Script**.
- When done click **Save**.
- If editing an existing scanner, select a disabled scanner and set the related fields the same way its mentioned above, and click **Save**.

2. Select the scanner that you have just created or edited, click **Edit**,and check to select **Use Site RunAs Secret**.
3. Click **Save**.

> If no RunAs secret is set on the site, you will get an error message when you try to save.

Developer Resources

This topic is a one-stop resource for Secret Server developers. It points to topics, as well as legacy knowledgebase articles. See the main "APIs and Scripting" on page 1466 section too.

Custom Reports

- "Creating a Custom Report" on page 887
- "Using Dynamic Parameters in Reports" on page 899

General Scripting

- "Configuring CredSSP for WinRM with PowerShell" on page 1479
- "Developer Resources" above
- "Creating and Using SSH Scripts" on page 1487
- "Creating and Using SQL Scripts" on page 1484
- "Creating RPC Scripts" on page 920
- "Using Secret Fields in Scripts" on page 1494
- "Using Webservices with IWA via PowerShell" on page 393

REST API

- "REST API PowerShell Scripts" on page 1503
- "REST API Reference Download" on page 1500

Scripting Dependencies

- "Change SQL Service Account Passwords without Restarting the SQL Service" on page 177
- "Dependency Token List" on page 1490
- "Discovery Error Messages" on page 559

Scripting Tools and CLI

- "Downloads for the Secret Server SDK for DevOps" on page 1551
- "Using the Secret Server SDK for DevOps" on page 1533
- Delinea Community GitHub Repository

SOAP API

- SOAP Web Services API Guide (Legacy feature)

Scripting Overview

Scripting in Secret Server is a powerful feature that allows administrators and DevOps teams to automate complex tasks, integrate with third-party applications, and enhance security processes. Secret Server supports both SOAP and RESTful APIs, enabling seamless interaction with various programming languages such as .NET, Java, Python, Ruby, and PowerShell.

Key Features

- Automating complex tasks
- Integrating with third-party applications
- Enhancing security processes

The Secret Server Software Development Kit (SDK) provides a secure way to access the API directly from the command line, eliminating the need to hard-code credentials into scripts. Additionally, PowerShell scripts can be used for custom password changers, dependency management, and integration with ticketing systems. By leveraging these scripting capabilities, organizations can ensure efficient and secure management of privileged accounts and secrets.

Key Resources

- PowerShell Scripts: Discover how to create and use PowerShell scripts to automate tasks and integrate with Secret Server.
- Creating and Using SQL Scripts: Learn how to leverage SQL scripts for managing dependencies and automating database interactions.
- Creating and Using SSH Scripts: Explore the use of SSH scripts for secure remote operations and automation.
- Dependency Tokens: Understand how to use dependency tokens in scripts to manage and update secret dependencies efficiently.
- Using Secret Fields in Scripts: Find out how to access and manipulate secret fields within your scripts for enhanced functionality.

PowerShell Scripts Overview

PowerShell scripting in Secret Server provides a versatile and powerful way to automate tasks, manage dependencies, and integrate with various systems. By using PowerShell scripts, administrators can enhance security processes and streamline operations within their IT environments.

Key Features

- Automating complex tasks
- Managing dependencies
- Integrating with third-party applications

PowerShell scripts can be used for custom password changers, dependency management, and integration with ticketing systems. By leveraging these scripting capabilities, organizations can ensure efficient and secure management of privileged accounts and secrets.

Key Resources

Creating and Using PowerShell Scripts: Explore detailed guides on creating and utilizing PowerShell scripts to automate tasks and integrate with Secret Server.

Secret RPC Scripts: Learn advanced techniques for using PowerShell scripts to manage complex dependencies and enhance security processes.

Creating and Using PowerShell Scripts

Overview

You can use PowerShell scripts in Secret Server to automate specific tasks. These scripts are useful in several places in Secret Server, such as in creating custom remote password changers, custom dependency changers, discovery scanners, and custom ticket system integration.

Creating a PowerShell Script

1. Develop your script. See:
   - "REST API PowerShell Scripts" on page 1503
   - "Using Secret Fields in Scripts" on page 1494
   - "Dependency Token List" on page 1490

   Do not edit the script with Windows Notepad. Instead, use Notepad++, Visual Studio Code, or Atom. Windows Notepad can add invisible characters that can cause issues.

   Using PowerShell in Secret Server involves passing a parameter string to the script. This string can contain literal values as well as tokens that represent values on the object in Secret Server that the script is attached to. For example, when creating a custom password changer, you pass in values such as the user name, old password, and new password using tokens that represent these values for whichever secret is running the password change script. Similarly, dependencies have a set of tokens that represent values on the dependency and its associated secret.

2. Go to **Admin > Scripts: PowerShell, SQL, SSH**, and on the Scripts page, click **Create Script**.

> In some older versions of Secret Server, you can find the **Scripts** option under **Remote Password Changing** on the **Administration** menu bar.

3. In the New Script page, type the script name in the **Name** text box.
4. Type a description in the **Description** text box.
5. Choose to select the **Enabled** checkbox.
6. In the **Script Type** field select **PowerShell**.
7. Click the **Category** dropdown list to select the type of script. This will determine where the script resides in Secret Server and more. For instance, the Dependency choice ensures that Remote Password Changing is turned on by enabling it on the Remote Password Changing page.
8. Paste your script into the **Script** text box.
9. When done, click **Save**. The new script appears in the table on the Scripts page.

Best Practices

Debugging

Scripts may contain debug lines to help you test the script. Debug statements use the `Write-Debug` command. For example: `Write-Debug "The users name is $Username"`

Script Arguments

There are input boxes for specifying arguments in places where the PowerShell scripts are used. Argument values are specified on a single line separated by a space. Values containing spaces should be enclosed in quotes. Parameters to PowerShell scripts are referenced through the zero-based `Args` array. It is often beneficial to assign `Args` variables to other more-meaningful named variables. For example:

Arguments: "Welcome back" and "Joe"

Script:

```
$greeting = $Args[0]
$name = $Args[1]
Write-Debug "$greeting $name"
```

Output: "Welcome back Joe"

Exceptions

In situations where the script should fail, given a specific set of conditions, an exception should be explicitly thrown. When an exception is thrown, the script stops running and the failure is logged in the system log. The script is considered to have successfully run if no errors or exceptions occur while processing. For example:

```
if ($meetsCondition -eq $false)
```

```
{
  throw "Did not meet condition"
}
```

Secret RPC Scripts

You can pass values calculated with a PowerShell script. For example, you can cycle an AWS IAM token. When changed within a PowerShell script, the new token is returned from the AWS API so it may be returned as a PSObject that will be used to update the secret.

DataItems

These return values from the script are called update "DataItems." These dataitems override any existing values and bypass any next values, including Auto-change Next Password. PowerShell RPC requires that you use a privileged account if a password is one of the items passed back in the DataItems.

Example returning a DataItems for a secret:

```
#at the end of the script add this return
$dataItem = New-Object –TypeName PSObject;$dataItem | Add-Member -MemberType NoteProperty -Name "Notes" -Value "NewValue1";$dataItem | Add-Member -MemberType NoteProperty -Name "Password" -Value " NewValue2";$dataItem | Add-Member -MemberType NoteProperty -Name "Machine" -Value " NewValue3";
return $dataItem
```

In this example, the fields “Notes”, “Password”, and “Machine” on the secret are updated with a “NewValue”. In your environment, these values are typically in relation to items that were generated during the RPC.

Dependencies

PowerShell dependencies can also return DataItems that are used to update a dependency.

This requires:

- Using a dependency changer with a defined scan-item template.
- The values passed back must be scan-item template fields. There are three built-in fields: "Description," "ServiceName," and "Machine." Advanced PowerShell dependency changers that use postscripts may also pass values between themselves.

Here is an example of returning a DataItem for a dependency:

```
# At the end of the script add this return
 $dataItem = New-Object –TypeName PSObject;$dataItem | Add-Member -MemberType NoteProperty -Name "Description" -Value "NewValue1";$dataItem | Add-Member -MemberType NoteProperty -Name "ServiceName" -Value " NewValue2";$dataItem | Add-Member -MemberType NoteProperty -Name "Machine" -Value " NewValue3";
return $dataItem
```

In this example, the values on the dependency are updated to match the “NewValue” that is being passed back from the PowerShell dependency changer.

Dependency Tokens

The available tokens are:

- `$UPDATED.<Token>` This give you the value the dependency script returned for the token DataItem. `<Token>` would be changed to match the field name returned that is needed in the script.
- `$SECRET.UPDATED.<Token>` This returns the value the secret password change script returned for the token DataItem.

Limitations

Files:

Files are supported but need to be returned as a string value in the DataItem. There could be encoding complications to watch out for as it uses UT8F.

Run Dependency:

- This loads the values from the current secret as DataItems.
- When running a dependency manually from the UI, which is typically done to update a dependency that was offline during the secret's password reset, DataItems that are saved on the secret as files are not supported, so the run will fail.

Test Dependency:

Tests done from the UI do not use DataItems, so they may return false positives with advanced dependencies.

Privileged Account:

A privilege account is required if changing the password of a secret.

Overview of WinRM with PowerShell

Windows Remote Management (WinRM) is a Microsoft technology that enables remote management of Windows systems over the network using various protocols and tools, including PowerShell. WinRM provides a standardized way to execute commands, transfer data, and perform remote administration tasks on Windows machines.

Using PowerShell scripts with WinRM offers several benefits and capabilities:

- Remote PowerShell Sessions: WinRM allows you to establish remote PowerShell sessions with one or more target computers. This enables you to run PowerShell commands and scripts on remote systems as if you were working directly on those machines.
- Remote Execution of Scripts: You can execute PowerShell scripts on remote computers without establishing an interactive session. This is useful for automating tasks or running scripts on multiple systems simultaneously.
- Secure Communication: WinRM supports various authentication methods, including Kerberos, Negotiate, and HTTPS/SSL, ensuring secure communication and data transfer between the local and remote systems.
- Credential Delegation: WinRM supports credential delegation, which allows you to run scripts or commands on remote systems using different credentials, enabling privilege escalation or impersonation when necessary.

- Session Configuration: WinRM provides session configurations that define the environmental settings, such as language mode, execution policies, and available modules, for remote PowerShell sessions. This ensures consistent and controlled execution environments across remote systems.
- PowerShell Remoting: PowerShell Remoting is a feature built on top of WinRM that simplifies the process of establishing remote PowerShell sessions and executing scripts or commands on remote systems.

To use PowerShell scripts with WinRM, you typically follow these steps:

1. Enable WinRM on the remote systems you want to manage.
2. Configure WinRM to allow remote PowerShell sessions and script execution.
3. Establish a remote PowerShell session using the Enter-PSSession or New-PSSession cmdlets, or execute scripts directly using the Invoke-Command cmdlet.
4. Run your PowerShell scripts on the remote systems within the established session or through direct script invocation.
5. Optionally, configure session configurations, authentication methods, and credential delegation as needed for your specific use case.

WinRM and PowerShell Remoting provide a powerful combination for remote management and automation of Windows systems. They enable IT administrators and system administrators to centrally manage and execute scripts, commands, and configurations across multiple remote machines, streamlining administrative tasks and improving operational efficiency.

Configuring CredSSP for WinRM with PowerShell

Introduction

In some cases, a PowerShell script may need to access resources outside of a Secret Server machine. This requires that any credentials are delegated to the target machine. Secret Server runs PowerShell scripts using Windows Remote Management (WinRM), which does not allow credential delegation by default. To allow credential delegation, the Secret Server machine must have Credential Security Support Provider (CredSSP) enabled. CredSSP is a security support provider that allows a client to delegate credentials to a target server.

CredSSP is required for dealing with double-hop scenarios. Per Microsoft documentation:

- CredSSP authentication is intended for environments where Kerberos delegation cannot be used.
- Support for CredSSP was added to allow a user to connect to a remote server and have the ability to access a second-hop machine, such as a file share.
- The first hop we make to their DE or web node (Enter-PSSession localhost), if they are running code locally from there and then doing a remote call (e.g. testing access to a remote UNC path) it will then require the second hop.

Enabling CredSSP for WinRM in Secret Server

1. Go to **Administration > Configuration search > Application**.
2. Click **Edit** at the top right.

3. Click to select the **Enable CredSSP Authentication for WinRM** checkbox.
4. Click the **Save** button.

> This is the global CredSSP settings and by default will configure CredSSP and connections to come *from* the Web server. This is used when **not using distributed engines.**

> If you are using distributed engines and you enable CredSSP at the site-specific level, these settings take precedence over this global CredSSP setting. Secrets will prioritize these site-specific settings. Therefore, if you plan on using CredSSP through a distributed engine, you should consider disabling the global setting seen below and only configure it at the site-specific level.

Configuring CredSSP for WinRM on the Secret Server Machine

1. Log on to the machine running Secret Server.
2. Run Windows PowerShell as an administrator.
3. Enable client-side CredSSP by running:

   ```
   Enable-WSManCredSSP -Role Client -DelegateComputer <Secret Server fully qualified machine name>
   ```

   For example:

   ```
   Enable-WSManCredSSP -Role Client -DelegateComputer <localhost>
   ```

   > localhost is the actual string that Secret Server uses to generate the PowerShell run space. Sometimes customers need both localhost and FQDN entries. *In theory*, those entries should be the same, thus not needing a second one.

4. Enable server-side CredSSP by running:

   ```
   Enable-WSManCredSSP -Role Server
   ```

5. The Web server always uses a specified account to run the PowerShell scripts. Considerations:
   - Ensure that account is added to the "Remote Management Users" local group on each Web server.
   - For RPCs with custom password changers, this would be "Change Password Using," and then select "Privileged Account."
   - For PowerShell password changers in the classic UI, this would be "Run PowerShell Using" and can alternatively be configured as the "Default Privileged Account" at the template level.
   - For custom dependencies using PowerShell scripts, this would be the "Run As" secret.
   - If you use any form of extensible discovery, this account needs to be the first secret that is linked to the scanner. Any additional secrets linked to the scanner are typically associated with authentication to the destination system.

Configuring CredSSP for WinRM on a Distributed Engine

You can alternatively configure CredSSP and the credential delegation to occur from your distributed engines by changing this setting at the site level:

1. Go to **Admin > Distributed Engine**. The Distribute Engine Configuration page appears.
2. Select the related site from the list.
3. Scroll down to see the **Enable CredSSP Authentication for WinRM** listing in the **Advanced Site Configuration** section.
4. If it is not enabled (value set to No), log on to each of your distributed engines where CredSSP is enabled.
5. Run Windows PowerShell as an administrator.
6. Enable client-side CredSSP by running:

   ```
   Enable-WSManCredSSP -Role Client -DelegateComputer <distributed engine fully qualified machine name>
   Enable-WSManCredSSP -Role Client -DelegateComputer <localhost>
   ```

   > localhost is the actual string that the Distributed Engine is using to generate the run space. Some customers need to have both the localhost and FQDN entry. In theory, both entries above should be the same, thus not needing a second entry.

7. Enable server-side CredSSP by running:

   ```
   Enable-WSManCredSSP -Role Server
   ```

8. The distribute engine will always use a specified account to run the PowerShell scripts. Considerations:
   - Ensure that account is added to the "Remote Management Users" local group on each engine where CredSSP is enabled.
   - For RPCs with custom password changers, this would be "Change Password Using," and then select "Privileged Account".
   - For PowerShell password changers in the classic UI, this would be "Run PowerShell Using" and can alternatively be configured as the "Default Privileged Account" at the template level.
   - For custom dependencies using PowerShell scripts, this would be the "Run As" secret.
   - If you use any form of extensible discovery, this account needs to be the first secret that is linked to the scanner. Any additional secrets linked to the scanner are typically associated with authentication to the destination system.
9. Ensure that the "Allow Delegating Fresh Credentials" group policy setting is enabled and is not disabled by a domain policy.
   a. Open the gpedit.msc file on your Secret Server machine or distributed engine, depending on where CredSSP is enabled
   b. Navigate to **Computer Settings > Administrative Templates > System > Credentials Delegation**.
   c. Edit the "Allow Delegating Fresh Credentials" setting.
   d. Verify that it is Enabled.
   e. Click "Show..."

f. Verify that the list contains an entry that begins with "wsman/" and ends with the fully qualified machine name of the Secret Server machine or distributed engine.

g. If destination systems are non-domain joined or on another domain without a trust, it may be required for you to add in an entry for **each** destination system you wish to run the script or do discovery on (as examples). Consider collecting a list of all destination FQDNs for your specific use case and adding them all in one go.

10. Depending on where CredSSP is configured (Web server or distributed engine), run the following commands:

    - View existing entries: `Get-Item WSMan:\localhost\Client\TrustedHosts`
    - Adding computers if your TrustedHosts list is empty: `Set-Item WSMan:\localhost\Client\TrustedHosts -Value <ComputerName>,[<ComputerName>]`
    - Adding computers to your existing TrustedHosts list:

      ```
      $curList = (Get-Item WSMan:\localhost\Client\TrustedHosts).value
      Set-Item WSMan:\localhost\Client\TrustedHosts -Value "$curList, Server01"
      ```

11. On the destination system, if it is on a separate domain without a trust or non-domain joined, add the reverse WSman entries so the destination system trusts either Secret Server or your engines. Run one of the following commands:

    Web server:

    ```
    Set-Item WSMan:\localhost\Client\TrustedHosts -Value <Web Server 1 FQDN>,[<Web Server 2 FQDN>]
    ```

    Engine:

    ```
    Set-Item WSMan:\localhost\Client\TrustedHosts -Value <Distributed Engine 1 FQDN>,
    [<Distributed Engine 2 FQDN>]
    ```

12. Restart either Secret Server or the engine you just trusted.

Enabling CredSSP on Secret Server Agents for PowerShell Script Dependencies

Remote agents were upgraded to distributed engines in Secret Server version 8.9. This section only applies to Secret Server versions 8.8.000020 and earlier.

Remote Agents are only needed for networks that are not directly connected to the network that Secret Server is installed on. If you are not using remote agents, disregard this section.

By default, Secret Server agents inherit the "Enable CredSSP Authentication for WinRM" setting from Secret Server; however, you can override this in the agent configuration file as follows:

1. On the machine running the agent, locate the agent program files. By default, they are at `C:\Program Files (x86)\Thycotic Software Ltd\Secret Server Agent`.
2. Edit the `SecretServerAgentService.exe.Config` file in a text editor.
3. Locate the "UnencryptedSettings" section.
4. Add a new key to that section for EnableCredSSPForWinRM and set it to true. For example:

   ```
   <add key="EnableCredSSPForWinRM" value="true" />
   ```

5. Restart the "Secret Server Agent" service to apply the setting.

Configuring WinRM for PowerShell

Overview

Secret Server relies on Windows Remote Management (WinRM) components to run PowerShell scripts. This requires configuration on the Secret Server Web server and any distributed engines. By default, Secret Server uses `http://localhost:5985/wsman` as the WinRM endpoint. The endpoint URI can be seen under **Admin > Configuration** or **Admin > Distributed Engine > Manage Sites > Local** if using distributed engines. At the moment, we only support running PowerShell scripts on localhost. If you are new to PowerShell Remoting please review PowerShell Remoting Security Considerations.

Configuration (Domain Joined)

Run PowerShell as an Administrator and execute:

```
Enable-PSRemoting
```

See Enable-PSRemoting for more information.

This command performs the following steps:

- Runs the Set-WSManQuickConfig cmdlet, which performs the following tasks:
  - Starts the WinRM service
  - Sets the startup type on the WinRM service to Automatic
  - Creates a listener to accept requests on any IP address
  - Enables a firewall exception for WS-Management communications
  - Registers the Microsoft.PowerShell and Microsoft.PowerShell.Workflow session configurations, if it they are not already registered
  - Registers the Microsoft.PowerShell32 session configuration on 64-bit computers, if it is not already registered
  - Enables all session configurations
  - Changes the security descriptor of all session configurations to allow remote access.
- Restarts the WinRM service to make the preceding changes effective.

Verifying Listeners

Confirm the listener:

```
Get-WSManInstance -ResourceURI winrm/config/listener -SelectorSet @
{Address="*";Transport="http"}
```

The output should reflect the newly configured listener with **Enabled : true**

Additional Considerations

> Ensure that the account running WinRM and connecting to the target machine has the permissions to access that machine over the network. Configure the policy value for **Computer Configuration > Windows Settings > Security Settings > Local Policies > User Rights Assignment > Access this computer from the network** to include the service account or user account that will need access to this computer over the network via WinRM.

- By default two BUILTIN groups are allowed to use PowerShell Remoting as of v4.0: Administrators and Remote Management Users.

  ```
  (Get-PSSessionConfiguration -Name Microsoft.PowerShell).Permission
  ```

- Sessions are launched by Secret Server under the user's context, which means all the same security controls and policies apply within the session.
- See Investigating PowerShell Attacks by FireEye for additional security considerations.
- Your environment may already be configured for WinRM. If your server is already configured for WinRM but it is not using the default configuration, you can change the URI to use a custom port or URLPrefix.

Configuration (Standalone)

By default WinRM uses Kerberos for authentication. Since Kerberos is not available on machines that are not joined to the domain, HTTPS is required for secured transport of the password. Only use this method if you are going to run scripts from a Secret Server Web server or distributed engine that is **not** joined to the domain.

> WinRM HTTPS requires a local computer "Server Authentication" certificate with a CN matching the hostname that is not expired, revoked, or self-signed. A certificate would need to be installed on each endpoint Secret Server or the engine manages.

Create the new listener:

```
New-WSManInstance - ResourceURI winrm/config/Listener -SelectorSet @{Transport=HTTPS} -
ValueSet @{Hostname="HOST";CertificateThumbprint="XXXXXXXXXX"}
```

Creating and Using SQL Scripts

You can use SQL scripts in Secret Server to automate specific tasks. You can configure a SQL script as a dependency of a secret and run after the password is successfully changed on the secret.

Creating a SQL Script

1. Navigate to **Admin > Scripts: PowerShell, SQL, SSH**, and on the Scripts page, click **Create Script**.
2. In the New Script page, type the script name in the **Name** text box.
3. Type a description in the **Description** text box.
4. Choose to select the **Enabled** checkbox.

5. In the **Script Type** field select **SQL**.
6. Click the **Category** dropdown list to select the type of script. This will determine where the script resides in Secret Server and more. For instance, the Dependency choice ensures that Remote Password Changing is turned on by enabling it on the Remote Password Changing page.
7. Select the related **Password Changer** from the dropdown.
8. Paste your script into the **Script** text box.
9. When done, click **Save**. The new script appears in the table on the Scripts page.

Using Parameters

Where SQL scripts are used, we provide an arguments text box. It is often beneficial to assign variables to other more meaningful variables.

Examples

SQL

```
UPDATE TABLE cmsuser
SET password = PWDENCRYPT(@Password) WHERE username = @Username;
```

MySQL

```
UPDATE TABLE cmsuser
SET password = PASSWORD(?) WHERE username = ?;
```

PostgreSQL

```
UPDATE TABLE cmsuser
SET passwd = CRYPT(?, GEN_SALT('sha256')) WHERE username = ?;
```

ODBC

```
UPDATE TABLE cmsuser
SET passwd = $Password WHERE username = $Username;
```

Basic Oracle PL/SQL

```
INSERT INTO cmuser values (:username, :password);
```

Advanced Oracle PL/SQL

```
EXECUTE IMMEDIATE 'alter user '|| :username ||' identified by "' || :password || '"';
```

Returning Errors

In situations where the script should fail given specific conditions, the script should explicitly throw an exception. When an exception is thrown, the script stops running and the failure is recorded in the system Log. The script is considered to have successfully run if no errors or exceptions occur while processing.

Examples

SQL

```
RAISERROR(N'ERROR: %s', 14, 1, N'Failure');
```

MySQL

```
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'ERROR: Failure';
```

PostgreSQL

```
DO language plpgsql $$
BEGIN
RAISE EXCEPTION 'ERROR (14)';
END
$$;
```

ODBC

```
RAISERROR(N'ERROR: %s', 14, 1, N'Failure');
```

SQL Example

An issue that occurs in SQL Server database environments is when a linked database is set up with a credential and that credential's password changes. To counter this, you can set up a SQL script to run as a dependency after the password change occurs to drop and recreate the link. Note that you may need to edit the option depending on the desired linked server configuration.

Example

SQL

```
EXEC master.dbo.sp_dropserver @server=@MACHINE, @droplogins='droplogins'
EXEC master.dbo.sp_addlinkedserver @server = @MACHINE, @srvproduct=N'SQL Server'
EXEC master.dbo.sp_addlinkedsrvlogin
@rmtsrvname=@MACHINE,@useself=N'True',@locallogin=NULL,@rmtuser=NULL,@rmtpassword=NULL
EXEC master.dbo.sp_addlinkedsrvlogin
@rmtsrvname=@MACHINE,@useself=N'False',@locallogin=@LOCALUSERNAME,@rmtuser=@REMOTEUSERNAME
,@rmtpassword=@PASSWORD
```

Parameters

- `@MACHINE` The machine or instance of the server where the linked database exists. For instance, `SERVER\SQL2014`.
- `@LOCALUSERNAME` The local login on the server where the linked database is configured.
- `@REMOTEUSERNAME` The username that is set in the linked database's security info for connecting to the linked database. This should be the username of the secret that the dependency is on.
- `@PASSWORD` This will be the new password after the SQL account's password is changed.

Creating and Using SSH Scripts

SSH scripts can be used in Secret Server to automate specific tasks. An SSH script can be configured as a dependency of a secret and run after the password is successfully changed on the secret.

Creating an SSH Script

1. Navigate to **Admin > Scripts: PowerShell, SQL, SSH**, and on the Scripts page, click **Create Script**.
2. In the New Script page, type the script name in the **Name** text box.
3. Type a description in the **Description** text box.

4. Choose to select the **Enabled** checkbox.
5. In the **Script Type** field select **SSH**.
6. Click the **Category** dropdown list to select the type of script. This will determine where the script resides in Secret Server and more. For instance, the Dependency choice ensures that Remote Password Changing is turned on by enabling it on the Remote Password Changing page.
7. Paste your script into the **Script** text box.
8. When done, click **Save**. The new script appears in the table on the Scripts page.

Scripts return all output results, which allows you to debug your script using the echo command (or any command that will write to standard output).

For example:

```
echo The users name is $USERNAME
```

An input box for specifying arguments will exist in places where the SSH scripts are used. Argument values are specified next to their associated argument name. Parameters to SSH scripts are referenced through named variables. It is often beneficial to assign parameter variables to other more meaningful variables.

Another example:

```
$GREETING = Welcome back
$NAME = Joe
echo $GREETING $NAME
```

Output: `Welcome back, Joe`

In situations where the script should fail given a specific set of conditions, exit the script with a non-zero status code. When scripts exit with a non-zero status, the failure is logged in the System Log. The script is considered to have successfully run if it does not exit with a non-zero status code.

Example:

```
if [ ! $MEETS_CONDITION -eq 0 ]; then
echo Failure
exit 1;
fi
echo Success
```

Use Case

Update a user's password in the Apache user configuration file when it is changed in Secret Server:

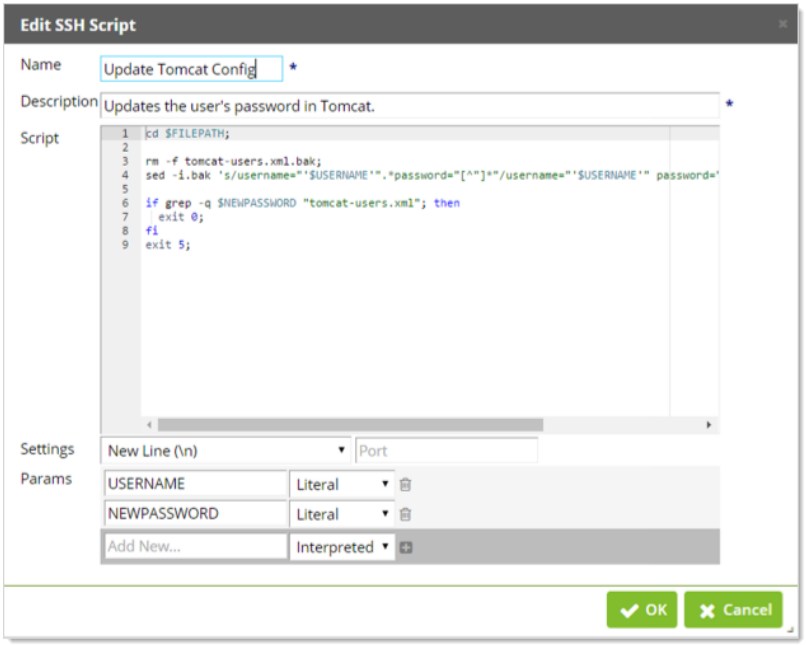
```
cd $FILEPATH;
rm -f tomcat-users.xml.bak;
sed -i.bak 's/username="'$USERNAME'".*password="[^"]*"/username="'$USERNAME'"
password="'$NEWPASSWORD'"/' tomcat-users.xml;
if grep -q $NEWPASSWORD "tomcat-users.xml"; then
exit 0;
fi
```

```
exit 5;
```

Adding an SSH Script as a Dependency

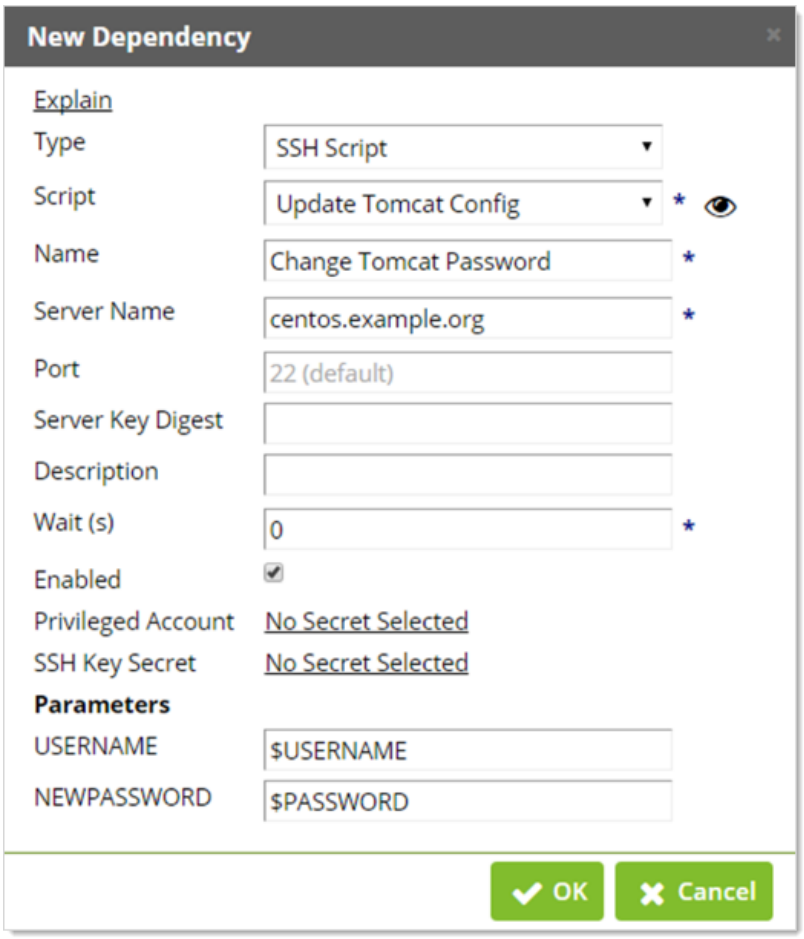
1. Navigate to **Admin > Scripts: PowerShell, SQL, SSH**, and on the Scripts page, click **Create Script**.
2. Fill out the required fields **Name**, **Description**, and **Script Type - SSH**.
3. Select **Category** as **Dependency**.
4. Paste your script into the **Script** text box.
5. (Optional) Change the setting for the environmental line ending in case your environment experiences issues with line termination.
6. (Optional) Specify the port that the script will connect to.

When filling out the script there is some important information to remember. If you would like to use one of the Secret Server variables such as $USERNAME, $NEWPASSWORD, or $CURRENTPASSWORD, you need to specify that they are parameters. For example, take the following script from our use case:



Within the script, parameters are specified with the dollar sign prefixed to the name. In the list of parameters, you simply specify their name without the dollar sign. This is so you can fill their values in later when implementing the script as a secret dependency.

When adding the SSH Script as a dependency to a secret, specify the built-in Secret Server variables with the dollar-sign prefix. For example, using our use case:



If a parameter in the script is not specified in the list of parameters, then they will use environmental variables on the remote host.

Dependency Token List

Overview

A list of all valid tokens that can be used in PowerShell, SSH, and SQL dependency script arguments.

PowerShell, SSH, and SQL dependencies can have script arguments that derive their values from values on the dependency, the secret it belongs to, or any other secrets associated for remote password changing. Tokens can also be used in ODBC connection string arguments. Script arguments are defined on dependency changers in Secret Server and on the dependency in earlier versions of Secret Server. The following table lists the tokens available to dependency scripts.

Best Practices

Do not use quotation marks (") in passwords for PowerShell scripts for RPC. Quotation characters are dropped from passwords when passing the $PASSWORD token to a custom password changer PowerShell script, resulting in an authentication failure.

You can add both $PRIOR... or $CURRENT... tags to any of the fields including custom fields.

Wrap Fields in Quotation Marks

When specifying PowerShell arguments, the $fieldname argument is replaced with the information in the field, and *then* the whole string is passed to PowerShell. Thus, if the string returned has spaces, PowerShell parses the spaces as separate arguments, which causes errors.

Therefore, we recommend you wrap your fields in two quotation marks. For example: `"$fieldname"`, which is then parsed by PowerShell as one string.

Avoid Using Spaces in Field Names

We recommend not using spaces within your field names. In many cases, that would not cause a problem, but if the first word in the field name is also a reserved word, it will cause an error because the reserved word is parsed separately from the rest of the field name.

For example, a field named:

`$machine name`

Because `$machine` is a reserved word, the parser would separate the reserved `$machine` value, followed by `name` as a separate argument.

Surrounding the field name in quotation marks would *not* fix this–Secret Server makes the substitution prior to PowerShell parsing the string.

Token List

Table: Dependency Tokens

Token	Available In	Translates To
$[x]${name of any field on secret}	pre-10.0	The value of the field with the same name on the nth secret on the RPC tab for use in custom password changing commands and scripts.
$[x]$PASSWORD	pre-10.0	The password on the nth secret on the RPC tab for use in custom password changing commands and scripts.
$[x]$USERNAME	pre-10.0	The username on the nth secret on the RPC tab for use in custom password changing commands and scripts. Example: $[1]$USERNAME is the username of the first associated secret.

Token	Available In	Translates To
${name of any field on secret}	pre-10.0	The value of the field on the secret with the same name. Examples: $DOMAIN matches the secret's "Domain" field, and $NOTES matches the "Notes" field.
${scan item field name} **Note:** The scan item token is designated for discovery purposes only, whereas other items may function in other capacities.	10.0	The name of any scan item field (defined on the scan item template) that is visible in the dependency edit dialog. If a scan item field is derived from a parent field, you may also use the parent field name as a parameter that translates to this field's value.
$CURRENTPASSWORD	pre-10.0	Deprecated. The password currently on the secret. $CURRENTPASSWORD is not available to the dependency script because when it runs the password has already changed. Use $PRIORPASSWORD instead.
$CURRENTPUBLICKEY	10.2	The public key currently on the secret (context-sensitive to whether script is run before or after key rotation).
$DATABASE	pre-10.0	The value of the Database field from the dependency. Only valid for SQL dependencies unless added as a field by the scan item template (see below).
$DEPENDENCYPRIVILEGEDPASSPHRASE	10.3	The private key passphrase on the privileged account assigned to the dependency.
$DEPENDENCYPRIVILEGEDPASSWORD	10.3	The password on the privileged account assigned to the dependency.
$DEPENDENCYPRIVILEGEDPRIVATEKEY	10.3	The private key on the privileged account assigned to the dependency.
$DEPENDENCYPRIVILEGEDUSERNAME	10.3	The user name on the privileged account assigned to the dependency.
$DEPENDENCYSSHKEY	10.3	The new SSH key to set on the dependency.
$DEPENDENCYSSHKEYPASSPHRASE	10.3	The new passphrase of the SSH key to be set on the dependency.

Token	Available In	Translates To
$MACHINE	10.0	The value of the Machine Name field on the dependency. This is always the second part of the dependency title (<service name> on <machine>).
$NEWPASSWORD	10.2	Deprecated. This is not available to the dependency at the time of the password change. Use $PASSWORD instead.
$NEWPUBLICKEY	10.2	The new public key that is being set on the secret.
$NEWACCESSKEY	10.2	The newly generated Access Key returned from an AWS IAM Key Rotation.
$NEWSECRETKEY	10.2	The newly generated Secret Key returned from an AWS IAM Key Rotation.
$PASSPHRASE	10.2	The passphrase used to encrypt the private key in a public/private key pair on this secret.
$PASSWORD	pre-10.0	The password on the secret.
$PORT	pre-10.0	The value of the Port field from the dependency. Only valid for SQL and SSH dependencies unless added as a field by the scan item template (see below).
$PRIORPASSPHRASE	10.2	The passphrase that was set on the secret before the current passphrase rotation.
$PRIORPASSWORD	10.2	In the context of a password change, this is the password at the beginning of the password change. The password that was set on the secret before the current password change. In the context of run dependency from the Dependency tab, this is the prior value of the password before the last password change.
$PRIORPUBLICKEY	10.2	The public key that was set on the secret before the current key rotation.
$PUBLICKEY	10.2	The public key on the secret.

Token	Available In	Translates To
$SERVICENAME	10.0	The value of the Service Name field on the dependency. Service Name may have a different name based on the dependency type, but it is always the first part of the dependency title (<service name> on <machine>).
$USERNAME	pre-10.0	The username on the secret.

The mappings of these parameters come from the remote password changer, so you can check the mapping by editing the secret template and selecting configure password changing.

Some of these tokens, such as `$PASSWORD`, `$CURRENTPASSWORD`, `$NEWPASSWORD`, and `$PRIORPASSWORD`, may seem to duplicate each other, but they have distinctions based on the context as described above.

In some cases, whether or not the dependency is being changed locally or through a distributed engine may have an impact on what these tokens return. This is due to the asynchronous nature of distributed engines. The newer tokens, such as `$NEWPASSWORD` and `$PRIORPASSWORD`, were created to address this issue. If you are using older tokens, such as `$PASSWORD` and `$CURRENTPASSWORD`, and seeing unexpected results, try using `$PRIORPASSWORD` and `$NEWPASSWORD`.

Using Secret Fields in Scripts

Secret Server supports using PowerShell, SSH, and SQL scripts as dependencies on a secret. These scripts can use information on the secret through the field name prepended with a `$`. For example, `$DOMAIN`, `$PASSWORD`, or `$USERNAME`. Linked secrets are accessible by `$[1]$FIELDNAME` for the first linked secret, `$[2]$FIELDNAME` for the second, and so on.

There are two contexts in which script dependencies run:

- As part of the RPC process. See "Creating RPC Scripts" on page 920.
- When run manually from the Dependencies tab on the secret.

For a complete list of tokens that are available to script dependencies, see "Dependency Token List" on page 1490.

REST API Overview

The REST API for Secret Server provides a robust and flexible way to programmatically interact with the Secret Server platform, enabling seamless integration with other IT systems and automation of critical security processes. The API supports both SOAP and RESTful interfaces, making it accessible to a wide range of programming languages and platforms, including .NET, Java, Python, Ruby, and PowerShell.

Key Functions

- Authenticating to Secret Server
- Creating and managing secrets
- Configuring folders and permissions
- Searching for secrets

The REST API uses token-based authentication to ensure secure access and supports various operations to manage secrets and their attributes. This capability is essential for organizations looking to integrate privileged account management into their existing workflows and enhance their DevOps practices.

Key Resources

- REST API Client Generation with Swagger: Learn how to generate clients for different programming languages using Swagger, which can help streamline your API interactions.
- REST API Reference Download: Access comprehensive documentation on REST API endpoints and parameters, essential for developers looking to integrate with Secret Server.
- REST API PowerShell Examples: Explore practical examples of using PowerShell scripts with the REST API, useful for automating tasks and enhancing your DevOps practices.

REST API Client Generation with Swagger

Secret Server contains an OpenAPI Swagger specification file that describes the REST API endpoints. There are several Swagger-based tools available to generate clients. This document describes using some popular ones to generate clients for different languages including C#, Java, and PowerShell. The generated client facilitates calling the Secret Server REST API and indicates API changes with new Secret Server versions.


Please use the article quick links on the right to jump to the section you are interested in.

Generating Clients

C# Client Using NSwagStudio

1. Download and install NSwagStudio (https://github.com/RicoSuter/NSwag/wiki/NSwagStudio)
2. Copy and paste the `swagger.json` file for "Documentation for token authentication." This is located at: `{Your SecretServer Base URL}/Documents/restapi/OAuth/swagger.json`
3. Suggested settings:
   - Namespace: SecretServerAuthentication
   - Inject HttpClient via constructor
   - Generate default values for properties
4. Generate output.
5. Copy output into a C# file in solution.

6. Copy and paste the swagger.json for "Documentation for REST API using bearer token authentication." It is located at `{Your SecretServer Base URL}/Documents/restapi/TokenAuth/swagger.json`.
7. Suggested Settings:
   - Namespace: SecretServerRestClient
   - Inject HttpClient via constructor
   - Generate default values for properties
8. Generate output.
9. Copy output into a C# file in solution

   See "Script Authentication Using Tokens" on page 1468 for creating a token to use in the examples below.

```
// Authenticate:
var httpClient = new HttpClient();
var token = "<TOKEN>";
// Set credentials (token):
httpClient.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue
("Bearer", token);
// Call REST API:
var client = new SecretServerRestClient.SecretsServiceClient
("https://secretserver.url.local/ss/api/v1", httpClient);
var search = client.SearchAsync(sortBy0_name: "lastHeartBeatStatus", sortBy0_
direction:"asc");
var results = search.Result;
```

C# or .NET Core Client Using OpenAPI Generator

> These client-generation instructions were written with OpenAPI Generator version 5.1.1, which was the latest version at the time. Future versions may fix issues that necessitated some workarounds. If you are using a newer version, you may need to make adjustments.

> These instructions assume that you have Java and .NET Core installed on your machine. They should work on all systems, but the syntax will need to be tweaked for other shells.

1. Follow the steps in Getting OpenAPI Generator.
2. Store the path to the relevant `swagger.json` files as variables. These variables can be either a local file path or a URL.

   ```
   $oauthSwagger = 'OAuth/swagger.json'$tokenAuthSwagger = 'TokenAuth/swagger.json'
   ```

3. Run the following commands to generate the necessary clients. You can run `java -jar openapi-generator-cli.jar help generate` to see advanced options that you may wish to configure. Options specific to the .NET Core generator are described on Config Options for csharp-netcore page.

   ```
   java -jar openapi-generator-cli.jar generate -i $oauthSwagger -g csharp-netcore --skip-
   validate-spec --package-name SecretServerOAuth --remove-operation-id-prefix --additional-
   properties=netCoreProjectFile=true -o oauth-csharpjava -jar openapi-generator-cli.jar
   generate -i $tokenAuthSwagger -g csharp-netcore --skip-validate-spec --package-name
   ```

```
SecretServerTokenAuth --remove-operation-id-prefix --additional-
properties=netCoreProjectFile=true -o tokenauth-csharp
```

4. Build the packages in release mode:

   ```
   dotnet build -c Release oauth-csharp/src/SecretServerOAuthdotnet publish -c Release
   tokenauth-csharp/src/SecretServerTokenAuth
   ```

5. Add the necessary .dll files to your project. The DLLs are located at the paths below:

   ```
   oauth-
   csharp/src/SecretServerOAuth/bin/Release/netstandard2.0/SecretServerOAuth.dlltokenauth-
   csharp/src/SecretServerTokenAuth/bin/Release/netstandard2.0/publish/*.dll
   ```

6. Test the API. For example:

   See "Script Authentication Using Tokens" on page 1468 for creating a token to use in the examples below.

   ```
   var token = "<TOKEN>";
   // Do not use tokenAuthConfig.AccessToken
   // This method can be called repeatedly as the access token expires, old values will be
   overwritten
   tokenAuthConfig.AddApiKey("Authorization", $"{token.TokenType} {token.AccessToken}");
   var foldersApi = new FoldersApi(tokenAuthConfig);
   var folder = await foldersApi.GetAsync(11);
   Console.WriteLine(folder);Console.ReadKey();
   ```

Java Client Using OpenAPI Generator

> These client-generation instructions were written with OpenAPI Generator version 5.1.1, which was the latest version at the time. If you are using a newer version, you may need to make adjustments.

> These instructions assume that you have Java configured on your machine. JDK 8 or greater is required. These instructions should work on all systems, but the syntax will need to be tweaked for other shells.

1. Follow the steps in Getting OpenAPI Generator.
2. Store the path to the relevant `swagger.json` files as variables. These variables can be either a local file path or a URL.

   ```
   $oauthSwagger = 'OAuth/swagger.json'$tokenAuthSwagger = 'TokenAuth/swagger.json'
   ```

3. Run the following commands to generate the necessary clients. You can run `java -jar openapi-generator-cli.jar help generate` to see advanced options that you may wish to configure. Options specific to the Java generator are described on the Config Options for java page.

   ```
   java -jar openapi-generator-cli.jar generate -i $oauthSwagger -g java --skip-validate-spec
   --invoker-package secretserver.oauth.client --api-package secretserver.oauth.api --model-
   package secretserver.oauth.model --group-id secretserver --artifact-id secretserver.oauth
   --remove-operation-id-prefix -o oauth-javajava -jar openapi-generator-cli.jar generate -i
   $tokenAuthSwagger -g java --skip-validate-spec --invoker-package
   secretserver.tokenauth.client --api-package secretserver.tokenauth.api --model-package
   secretserver.tokenauth.model --group-id secretserver --artifact-id secretserver.tokenauth
   --remove-operation-id-prefix -o tokenauth-java
   ```

4. Follow the instructions in `oauth-java/README.md` to include the package in your build process. Alternatively, follow the steps below to generate JAR files and test.

   ```
   cd oauth-java mvn clean install -DskipTestscd ..\tokenauth-java mvn clean install -DskipTests
   ```

5. Include the following JARs in your project (replace version numbers as necessary):

   ```
   oauth-java/target/secretserver.oauth-10.9.9.jartokenauth-java/target/secretserver.tokenauth-10.9.9.jartokenauth-java/target/lib/*.jar
   ```

6. Test the JAR files:

   See "Script Authentication Using Tokens" on page 1468 for creating a token to use in the examples below.

   ```
   // set to the root of the <Secret Server URL /> instance with no trailing slash
   String basePath = "https://thycotic.com/SecretServer";
   secretserver.oauth.client.ApiClient oauthClient =
   secretserver.oauth.client.Configuration.getDefaultApiClient();
   secretserver.tokenauth.client.ApiClient tokenAuthClient =
   secretserver.tokenauth.client.Configuration.getDefaultApiClient();
   String token = "<TOKEN>";tokenAuthClient.setBasePath(basePath
   + "/api");tokenAuthClient.setApiKey(tokenResponse.getTokenType() + " " + token);
   FoldersApi foldersApi = new FoldersApi(tokenAuthClient);
   FolderModel folderModel = foldersApi.get(11, false, false);
   System.out.println(folderModel);
   ```

PowerShell Using OpenAPI Generator

> These client-generation instructions were written with OpenAPI Generator version 5.1.1, which was the latest version at the time. If you are using a newer version, you may need to make adjustments. The PowerShell generator is in beta.

1. Follow the steps in Getting OpenAPI Generator.
2. Store the path to the relevant `swagger.json` files as variables. These variables can be either a local file path or a URL.

   ```
   $oauthSwagger = 'OAuth/swagger.json'$tokenAuthSwagger = 'TokenAuth/swagger.json\'
   ```

3. Run the following commands to generate the necessary clients. You can run `java -jar openapi-generator-cli.jar help generate` to see advanced options that you may wish to configure. Options specific to the PowerShell generator are described on Config Options for PowerShell page.

   > These commands set disallowAdditionalPropertiesIfNotPresent to false to work around a bug in the generator.

   ```
   java -jar openapi-generator-cli.jar generate -i $oauthSwagger -g PowerShell --skip-validate-spec --package-name SecretServerOAuth --remove-operation-id-prefix -o oauth-psjava -jar openapi-generator-cli.jar generate -i $tokenAuthSwagger -g PowerShell --skip-validate-spec --package-name SecretServerTokenAuth --remove-operation-id-prefix -o tokenauth-ps
   ```

4. Perform a find/replace to work around a bug in the generator:

```
Get-ChildItem ".\tokenauth-ps\src\SecretServerTokenAuth\Model\*.ps1" | Foreach-Object {
(Get-Content $_) | Foreach-Object { $_ -replace "\$AllProperties = \(\)", "$AllProperties
= @()" } | Set-Content $_ }
```

5. Run the following commands to build the PowerShell modules. -Verbose can be specified if desired.

   ```
   .\oauth-ps\Build.ps1.\tokenauth-ps\Build.ps1
   ```

6. Import both modules:

   ```
   Import-Module .\oauth-ps\src\SecretServerOAuthImport-Module .\tokenauth-
   ps\src\SecretServerTokenAuth
   ```

7. Set the base URL of each module. Only edit the first line below. The value must be the full URL to the root of Secret Server **with no trailing slash**, for example `https://thycotic.com/SecretServer`.

   ```
   $baseUrl = "https://thycotic.com/SecretServer"SecretServerOAuth\Set-Configuration -BaseUrl
   $baseUrlSecretServerTokenAuth\Set-Configuration -BaseUrl $baseUrl
   ```

8. Store a token - See "Script Authentication Using Tokens" on page 1468 for creating a token.

9. Set the TokenAuth module to use the token: `SecretServerTokenAuth\Set-Configuration -DefaultHeaders @{ Authorization = $token.token_type + " " + $token.access_token }`

10. Test the folder service Get endpoint (replace the ID as necessary):

    ```
    Get-FolderDetail -id 11
    ```

Postman Workspace

Postman is a REST client with a GUI to easily test API requests. To automatically generate a workspace with Secret Server API calls, click the **Import** button and choose whether to use a local file or a URL. Ensure the **Generate a Postman Collection** check box is selected. The program appears to be doing nothing while importing, but will inform you of success shortly.

Insomnia Workspace

Insomnia is a REST client similar to Postman. To automatically generate a workspace with Secret Server API calls, first click the arrow to open the dropdown, then click **Import/Export**. Click **Import Data**, then **From File** or **From URL** as desired.

Getting OpenAPI Generator

> These client-generation instructions were written with OpenAPI Generator version 4.3.0, which was the latest version at the time. Future versions may fix issues that necessitated some workarounds. If you are using a newer version, you may need to make adjustments.

OpenAPI Generator supports several installation methods, described on their CLI Installation page. You can pick the method most suitable for your environment. The commands in this article use the JAR method for simplicity, but other installation methods use mostly the same command format so only small tweaking should be required.

> All methods require the Java runtime (version 8+).

Self-Signed or Other Invalid Certificates

If you use an SSL certificate that is self-signed or otherwise not technically valid, OpenAPI Generator throws an error if you try to use a URL to `swagger.json` instead of a local file (or when using the Java client). To fix this, you need to import the certificate into Java's certificate store. The following example commands are for Windows, but the same concept applies to Mac and Linux as well.

The default password for the cacerts keystore is `changeit`.

Java 8 stores the cacerts keystore in JAVA_HOME unless it was explicitly changed. `keytool` should also be on your PATH if `java` itself is.

```
keytool -import -keystore "%JAVA_HOME%\jre\lib\security\cacerts" -file "path\to\cert.pem"
```

Java 11 provides a flag on `keytool` to find the cacerts keystore automatically.

```
keytool -import -cacerts -file "path\to\cert.pem"
```

Swagger 2.0 Notes

Some features in our REST API are not currently supported in Swagger 2.0. For example, the sort is an array of a complex objects. The actual API will accept multiple sort levels by passing this query string, `paging.sortBy[0].name=secretname&paging.sortBy[1].name=folderid`. The auto-generated client only allows you to specify the first sort by default. To specify multiple sorts, the serialization needs to be customized. We assumed that multiple sorting levels is probably an advanced feature and or choice will work for most.

REST API Reference Download

Overview

The Secret Server REST API guides are version specific. In fact, they are automatically generated when a Secret Server version is created. Thus, to ensure you have the correct guides *for* your Secret Server version, it is best (but not required) to access the documentation *from* that version of Secret Server; however there are many reasons why this might not be practical, so we provide download links below.

Accessing the Guides

To access the guides:

1. For Secret Server click the question mark icon in the top right of the dashboard and click **REST API Guide**. The Secret Server REST API Guide page appears.
2. Accessing Secret Server through the API and scripts can be powerful mechanism but ensure you are following best practices with the account and access for any script connecting to Secret Server. The Secret Server Software Development Kit for DevOps is recommended for all automated or machine-to-machine scripts. If not

using the SDK, creating an application account for the script access and limiting the permissions both from Role perspective and on Secret themselves is security best practice.

3. On the Secret Server REST API page of your instance, choose one of the three guides:
   - **Bearer token authentication:** Hyperlinked documentation for REST API access using token authentication.
   - **Token authentication:** Instructions for getting an authentication token.
   - **Windows Integrated Authentication:** Hyperlinked documentation for REST API access using Integrated Windows Authentication (IWA).

Downloading the Guides

> Only use the following links if you cannot easily access a guide from a version of Secret Server. There is a possibility the guides below are not the most current versions.

Current Version

Current Secret Server REST API Guide

Version Archive

- Secret Server 11.6.000003 REST API Guide
- Secret Server 11.5.000002 REST API Guide
- Secret Server 11.4.000031 REST API Guide
- Secret Server 11.4.000002 REST API Guide
- Secret Server 11.3.000003 REST API Guide
- Secret Server 11.1.000007 REST API Guide
- Secret Server 11.0.000008 REST API Guide
- Secret Server 11.0.000007 REST API Guide
- Secret Server 11.0.000006 REST API Guide
- Secret Server 10.9.000064 REST API Guide
- Secret Server 10.9.000033 REST API Guide
- Secret Server 10.8.000000 REST API Guide
- Secret Server 10.7.000000 REST API Guide
- Secret Server 10.6.000000 REST API Guide
- Secret Server 10.5.000000 REST API Guide
- Secret Server 10.4.000000 REST API Guide
- Secret Server 10.3.000000 REST API Guide
- Secret Server 10.2.000000 REST API Guide
- Secret Server 10.1.000000 REST API Guide

Understanding the Deprecation of V1

Version 1 (V1) of the Secret Server API has been deprecated and is no longer supported. While you might still be using some V1 endpoints, it's important to transition to V2 to take advantage of improved functionality and ongoing support. This change follows the OpenAPI standard, ensuring that updates do not disrupt your existing scripts by introducing new versions when necessary.

Key Differences You Should Know

- **Endpoint Changes**: In V2, some endpoints have equivalents in V1, but they often differ significantly. For example, the `/v1/secret-templates/{secrettemplateid}` endpoint has a V2 version that returns different data, excluding some fields that were available in V1. It's important to note that while V2 aims to optimize output, it may require adjustments to workflows to accommodate these changes.
- **Backward Compatibility Concerns**: While the goal is to maintain backward compatibility, there are instances where significant improvements necessitate changes. In hindsight, some endpoints might have been better introduced as entirely new endpoints rather than as part of V2, especially when they differ substantially from their V1 counterparts.
- **Output Optimization**: V2 endpoints are optimized to provide more relevant data, which might mean less information is returned. This can lead to a more efficient API, but you may need to adjust your workflows to accommodate these changes.
- **Field and Permission Adjustments**: V2 may include changes in field configurations and permissions, affecting how you access and manipulate data. While V1 outputs were extensive, V2 might require additional calls to retrieve specific field details.
- **New Functionalities**: V2 introduces new functionalities and endpoints that may not have direct equivalents in V1. Exploring these new features can enhance your use of the API.

Steps to Transition to V2

- **Review Documentation**: Make sure you have the latest API documentation, which provides clear guidance on the differences between V1 and V2 endpoints. This will help you understand the changes and how to implement them in your systems.
- **Identify Impacted Workflows**: Determine which of your workflows rely on V1 endpoints and assess how they will be affected by the transition to V2. You may need to rework certain processes to align with the new API structure.
- **Utilize New Endpoints**: Explore the new endpoints and functionalities in V2 that can enhance your existing workflows or introduce new capabilities.

Note on API Versioning

The transition to V2 is part of an ongoing effort to improve the API's efficiency and functionality. However, it's acknowledged that maintaining backward compatibility is crucial, and future changes will strive to adhere to this principle unless there is a compelling reason to deviate.

REST API Examples

The REST API examples for Secret Server provide practical demonstrations of how to interact with the Secret Server platform programmatically using various programming languages. These examples cover essential operations such as authenticating to Secret Server, retrieving secrets, updating secret fields, and managing secret attributes. The examples are designed to be user-friendly and leverage standard HTTP methods and status codes, making them accessible for developers of all skill levels. They include detailed guides for using Python, Perl, and PowerShell, showcasing how to perform common tasks and integrate Secret Server into custom workflows. By following these examples, developers can automate critical security processes, enhance their DevOps practices, and ensure secure and efficient management of privileged accounts.

- REST API PowerShell Scripts
- REST API Python Scripts
- REST API Perl Scripts

REST API PowerShell Scripts

To use the REST API you first must enable Webservices. To do so, go to Admin > Configuration general tab. Enabling Webservices simply makes the ASP.NET REST Webservices built into Secret Server available.

For a full reference of the REST endpoints and parameters, see the "REST API Reference Download" on page 1500.

When using the API to search secrets, the account used must have at least "view" permissions on the full folder path to find the secret.

Secret Server Cloud exclusively supports TLS 1.2. This version includes fixes for known vulnerabilities in older TLS versions and will eventually be required for PCI compliance. The following will need to be added to the top of your scripts because PowerShell defaults to TLS 1.0:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

Secret Server Cloud API users: If Delinea One is enabled, *and* an API user logging on has an email address set in Secret Server, the user *must* use the Delinea One password, instead of the local Secret Server password. Using a local account password will fail. Use one of these workarounds:

- Disable Delinea One, if not needed
- Use the Delinea One password instead
- Use a user account that does not have an email address in Secret Server

> Please see "Script Authentication Using Tokens" on page 1468 for instructions on acquiring a token for the scripts below.

Authentication

See "Script Authentication Using Tokens" on page 1468 for creating a token to use in the script examples below.

Searching Secrets

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    $filters = "?filter.includeRestricted=true&filter.searchtext=< mySearchText >"
    Write-Host "------------------------------"
    Write-Host "----- Secret Search Values -----"
    Write-Host "------------------------------"

#?filter.searchfield=username&filter.searchfield=displayname&filter.searchfield=filter.sea
rchText=mister&filter.includeinactive=true" -Headers $headers
    $result = Invoke-RestMethod "$api/secrets$filters" -Headers $headers
    Write-Host $result.filter.searchField
    Write-Host $result.total
    foreach($secret1 in $result.records)
    {
       Write-Host $secret1.id" - "$secret1.name" - "$secret1.folderId -
$secret1.lastHeartBeatStatus
    }
    Write-Host "------------------------------"
    Write-Host "----- Secret Lookup Values -----"
    Write-Host "------------------------------"

#?filter.searchfield=username&filter.searchfield=displayname&filter.searchfield=filter.sea
rchText=mister&filter.includeinactive=true" -Headers $headers
    $result = Invoke-RestMethod "$api/secrets/lookup$filters" -Headers $headers
    Write-Host $result.filter.searchField
    Write-Host $result.total
    foreach($secret in $result.records)
    {
       Write-Host $secret.id" - "$secret.value
    }
    Write-Host "------------------------------"
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
```

```
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Creating a Secret

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    #stub
    $templateId = <Your Secret Template ID>
    $secret = Invoke-RestMethod $api"/secrets/stub?filter.secrettemplateid=$templateId" -
Headers $headers
    #modify
    $secret.name = <Your Secret Name>
    $secret.secretTemplateId = $templateId
    $secret.AutoChangeEnabled = $false
    $secret.autoChangeNextPassword = <Next Password Value>
    $secret.SiteId = <Your Site ID>
    $secret.folderId = <Your Folder ID>
    foreach($item in $secret.items)
    {
      if($item.fieldName -eq "Domain")
      {
        $item.itemValue = <Your Domain>
      }
      if($item.fieldName -eq "Username")
      {
        $item.itemValue = <Username>
      }
      if($item.fieldName -eq "Password")
      {
        $item.itemValue = <Password>
      }
    }
    $secretArgs = $secret | ConvertTo-Json
    #create
    Write-Host ""
    Write-Host "-----Create secret -----"
    $secret = Invoke-RestMethod $api"/secrets/" -Method Post -Body $secretArgs -Headers
$headers -ContentType "application/json"
```

```
    $secret1 = $secret | ConvertTo-Json
    Write-Host $secret1
    Write-Host $secret.id
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host $responseBody
}
```

Editing a Secret

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    #get
    $secretId = <Secret ID>
    $secret = Invoke-RestMethod $api"/secrets/$secretId/" -Headers $headers
    #modify
    $secret.RequiresComment = $true #Example only. Available fields to edit can be found
at https://<Your URL>/RestApiDocs.ashx?doc=token-
help#tag/Secrets/operation/SecretsService_UpdateSecret
    $secretArgs = $secret | ConvertTo-Json
    #update
    Write-Host ""
    Write-Host "-----Update secret -----"
    $secret = Invoke-RestMethod $api"/secrets/$secretId" -Method Put -Body $secretArgs -
Headers $headers -ContentType "application/json"
    $secretUpdate = $secret | ConvertTo-Json
    Write-Host $secretUpdate
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
```

```
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host $responseBody
}
```

Checking in a Secret

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    $secretId = <Your Secret ID>
    $secret = Invoke-RestMethod $api"/secrets/$secretId/check-in" -Method Post -Body
$secretArgs -Headers $headers -ContentType "application/json"
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host $responseBody
}
```

Deleting a Secret

```
try
{
    $api = "<URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    $secretId = <Your Secret ID>
    Write-Host "----- Delete a Secret -----"
    $deletemodel = Invoke-RestMethod "$api/secrets/$secretId" -Headers $headers -Method
DELETE -ContentType "application/json"
    Write-Host $deletemodel
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
```

```
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Creating a User

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    # create user
    Write-Host ""
    Write-Host "----- Create a User -----"
    $userCreateArgs = @{ #More fields available at https://<Your
URL>/RestApiDocs.ashx?doc=token-help#tag/Users/operation/UsersService_CreateUser
        userName = <Username>
        password = <Password>
        DisplayName = <Display Name>
        enabled = $true #Not required. Default is false.
    } | ConvertTo-Json
    $user = Invoke-RestMethod "$api/users" -Headers $headers -Method Post -ContentType
"application/json" -Body $userCreateArgs
    Write-Host "New User ID : " $user.id
}
catch
{
    Write-Debug "----- Exception -----"
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
```

```
}
```

Update Secret Field

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    #Secret ID and Field to update
    $secretId = <Your Secret ID>
    $fieldToUpdate = <Field to Update>
    $endpoint = "$api/secrets/$secretId/fields/$fieldToUpdate"
    $body = @{
        value = <New Value>
    }| ConvertTo-Json
    echo $endpoint
    echo ----------------------------------------------------------------------------------------
------------------
    echo ----------------------------------------------------------------------------------------
------------------
    echo "Updating Field $fieldToUpdate"
    $response = Invoke-RestMethod -Method Put -Uri $endpoint -Headers $headers -
ContentType "application/json" -Body $body
    echo $response;
}
catch
{
    Write-Debug "----- Exception -----"
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Get Secret Field Value

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    #Secret ID and Field to test against
    $secretId = <Your Secret ID>
    $field = <Field name>
    $endpoint ="$api/secrets/$secretId/fields/$field"
    $response = $null
    $response = Invoke-RestMethod -Method Get -Uri $endpoint -Headers $headers
    echo $response;
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Upload File to Secret

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    #Secret ID and File to upload
    $fileSecretId = <Your Secret ID>
    $fileFieldToUpdate = <Field Name to Store File>
    echo ------------------------------------------------------------------------------
------------------
    echo "Uploading file from $fileFieldToUpdate"
    $endpoint ="$api/secrets/$fileSecretId/fields/$fileFieldToUpdate"
    echo $endpoint
    $secretArgs = @{
        fileName = <File Name>
```

```
        fileAttachment = [IO.File]::ReadAllBytes(<File Path>)
    } | ConvertTo-Json
    $response = $null
    $response = Invoke-RestMethod -Method Put -Uri $endpoint -Headers $headers -Body
$secretArgs -ContentType "application/json"
    echo $response
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Download File From Secret

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    #Secret ID and File to download
    $fileSecretId = <Your Secret ID>
    $fileFieldToUpdate = <Field Name Storing File>
    $downloadPath = <Your Download path, including file name being downloaded>
    echo "Downloading file from $fileFieldToUpdate"
    $endpoint ="$api/secrets/$fileSecretId/fields/$fileFieldToUpdate"
    echo $endpoint
    $response = $null
    $response = Invoke-RestMethod -Method Get -Uri $endpoint -Headers $headers -OutFile
$downloadPath
    Write-Host $response.Length
    Write-Host $response
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
```

```
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Expiring a Token

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    foreach($user in $pagedUsers.records)
    {
        Write-Host $user.userName
    }
    # expire token
    Write-Host ""
    Write-Host "----- Expire Token -----"
    $expireToken = Invoke-RestMethod "$api/oauth-expiration" -Headers $headers -Method
Post
    # This part should fail with a 403 Forbidden
    Write-Host ""
    Write-Host "----- Expect an error -----"
    $secrets = Invoke-RestMethod "$api/secrets" -Headers $headers
}
catch
{
    Write-Debug "----- Exception -----"
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd() | ConvertFrom-Json
    Write-Host  $responseBody.errorCode " - " $responseBody.message
```

```
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Add Folder

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    # Get Folder Stub
    $folderStub = Invoke-RestMethod $api"/folders/stub" -Method GET -Headers $headers -
ContentType "application/json"
    $folderStub.folderName = <Your Folder Name>
    $folderStub.folderTypeId = 1
    $folderStub.inheritPermissions = $false
    $folderStub.inheritSecretPolicy = $false
    $folderArgs = $folderStub | ConvertTo-Json
    $folderAddResult = Invoke-RestMethod $api"/folders" -Method POST -Body $folderArgs -
Headers $headers -ContentType "application/json"
    $folderId = $folderAddResult.id
    if($folderId-gt 1)
    {
        echo ""
        echo "------------------------"
        echo "--Add Folder Successful--"
        echo "------------------------"
        echo ""
        echo $folderAddResult | ConvertTo-Json
    }
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
```

```
        $modelState
    }
}
```

Delete Folder

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    # Get Folder Stub
    $folderStub = Invoke-RestMethod $api"/folders/stub" -Method GET -Headers $headers -
ContentType "application/json"
    $folderId = <Your Secret ID>
    $folderArgs = $folderStub | ConvertTo-Json
    $folderDelete = Invoke-RestMethod $api"/folders/$folderId" -Method DELETE -Body
$folderArgs -Headers $headers -ContentType "application/json"
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Get Folder

This script requires Secret Server 10.1 or later.

```
try
{
```

```
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    # Get Folder Stub
    $folderStub = Invoke-RestMethod $api"/folders/stub" -Method GET -Headers $headers -
ContentType "application/json"
    $folderId = <Your Secret ID>
    $folderGetResult = Invoke-RestMethod $api"/folders/$folderid" -Method GET -Headers
$headers -ContentType "application/json"
    if($folderGetResult.id -eq $folderId)
    {
        echo ""
        echo "-----------------------"
        echo "--Get Folder Successful--"
        echo "-----------------------"
        echo ""
        echo $folderGetResult | ConvertTo-Json
    }
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Add Child Folder

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    # Get Folder Stub
```

```
    $folderStub = Invoke-RestMethod $api"/folders/stub" -Method GET -Headers $headers -
ContentType "application/json"
    $folderStub.folderName = <Folder Name>
    $folderStub.folderTypeId = 1
    $folderStub.inheritPermissions = $false
    $folderStub.inheritSecretPolicy = $false
    $folderStub.parentFolderId = <Parent Folder ID>
    $folderArgs = $folderStub | ConvertTo-Json
    $folderChildAddResult = Invoke-RestMethod $api"/folders" -Method POST -Body
$folderArgs -Headers $headers -ContentType "application/json"
    $childfolderId = $folderChildAddResult.id
    if($childfolderId-gt 1)
    {
        echo ""
        echo "-----------------------"
        echo "--Add Child Folder Successful--"
        echo "-----------------------"
        echo ""
        echo $folderChildAddResult | ConvertTo-Json
    }
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Update Folder

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
```

```
    # Get Folder Stub
    $folderStub = Invoke-RestMethod "$api/folders/stub" -Method GET -Headers $headers -
ContentType "application/json"
    $folderId = <Folder ID of folder to update>
    $folderStub.folderName = <Folder Name>
    $folderStub.folderTypeId = 1
    $folderStub.id = $folderId
    $folderUpdateArgs = $folderStub | ConvertTo-Json
    $folderUpdateResult = Invoke-RestMethod "$api/folders/$folderId" -Method PUT -Body
$folderUpdateArgs -Headers $headers -ContentType "application/json"
    Write-Host $folderUpdateResult
    if($folderUpdateResult.folderId -eq $folderId)
    {
        echo ""
        echo "-----------------------"
        echo "--Update Folder Successful--"
        echo "-----------------------"
        echo ""
        echo $childFolderUpdateResult | ConvertTo-Json
    }
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Search Folders

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
```

```
    # Get Folder Stub
    $folderStub = Invoke-RestMethod $api"/folders/stub" -Method GET -Headers $headers -
ContentType "application/json"
    $searchFilter ="?filter.searchText=<Search Text>"
    $searchResults = Invoke-RestMethod $api"/folders$searchFilter" -Method GET -Headers
$headers -ContentType "application/json"
    $folder = $searchResults.records[0]
    echo $searchResults
    echo $folder
    $name =  <Folder Name>
    if($searchResults.total -gt 0 -and $folder.folderName -eq $name)
    {
        echo ""
        echo "-------------------------------"
        echo "--Search Folder Successful--"
        echo "-------------------------------"
        echo ""
        echo $group
    }
    else
    {
        Write-Error "ERROR: Failed to Search Folders."
        return
    }
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Lookup Folders

This script requires Secret Server 10.1 or later.

```
try
{
```

```
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    # Get Folder Stub
    $folderStub = Invoke-RestMethod $api"/folders/stub" -Method GET -Headers $headers -
ContentType "application/json"
    $lookupFilter = "?filter.searchText=<Search Text>"
    $lookupResults = Invoke-RestMethod $api"/folders/lookup$lookupFilter" -Method GET -
Headers $headers -ContentType "application/json"
    $folder = $lookupResults.records[0]
    echo $lookupResults
    echo $folder
   if($searchResults.total -gt 0 -and $folder.value -eq $name)
   {
       echo ""
       echo "------------------------------"
       echo "--Lookup Folder Successful--"
       echo "------------------------------"
       echo ""
       echo $folder
   }
   else
   {
       Write-Error "ERROR: Failed to Lookup Folders."
       return
   }
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Add Folder Permissions

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    # Get Folder Stub
    $folderStub = Invoke-RestMethod "$api/folders/stub" -Method GET -Headers $headers -
ContentType "application/json"
    $folderId = <Your Folder ID>
    $folderPermissionCreateArgs = Invoke-RestMethod $api"/folder-
permissions/stub?filter.folderId=$folderId" -Method GET -Headers $headers -ContentType
"application/json"
    $folderPermissionCreateArgs.GroupId = <Group ID. $null if assigning by user.>
    $folderPermissionCreateArgs.UserId = <User ID. $null if assigning by group.>
    $folderPermissionCreateArgs.FolderAccessRoleName = <Role Name>
    $folderPermissionCreateArgs.SecretAccessRoleName = <Role Name>
    $permissionArgs = $folderPermissionCreateArgs | ConvertTo-Json
    $permissionResults = Invoke-RestMethod "$api/folder-permissions" -Method POST -
Headers $headers -Body $permissionArgs -ContentType "application/json"
    if($permissionResults.FolderId -eq $folderId)
    {
        echo ""
        echo "---------------------------------------"
        echo "--Add Folder Permissions Successful--"
        echo "---------------------------------------"
        echo ""
        echo $permissionResults
    }
    else
    {
        Write-Error "ERROR: Failed to Add Folder Permissions."
        return
    }
    $folderPermissionId = $permissionResults.id
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

Delete Folder Permissions

This script requires Secret Server 10.1 or later.

```
try
{
    $api = "<Secret Server URL>/api/v1"
    $token = "<TOKEN>"
    $headers = New-Object "System.Collections.Generic.Dictionary[[String],[String]]"
    $headers.Add("Authorization", "Bearer $token")
    $folderId = <Folder ID of folder being modified>
    $userId = <User ID of the user being removed from folder>
    # Get Folder Stub
    $folderStub = Invoke-RestMethod "$api/folders/$folderId" -Method GET -Headers
$headers -ContentType "application/json"
    $folderPermissionId = Invoke-RestMethod "$api/folder-
permissions/$folderId/?filter.userId=$userId" -Method GET -Headers $headers -ContentType
"application/json"
    $permissionDeleteResult = Invoke-RestMethod "$api/folder-
permissions/$folderPermissionId" -Method DELETE -Headers $headers -ContentType
"application/json"
    if($permissionDeleteResult.id -eq $folderPermissionId)
    {
        echo ""
        echo "----------------------------------------"
        echo "--Remove Folder Permissions Successful--"
        echo "----------------------------------------"
        echo ""
    }
    else
    {
        Write-Error "ERROR: Failed to Remove Folder Permissions."
        return
    }
}
catch [System.Net.WebException]
{
    Write-Host "----- Exception -----"
    Write-Host  $_.Exception
    Write-Host  $_.Exception.Response.StatusCode
    Write-Host  $_.Exception.Response.StatusDescription
    $result = $_.Exception.Response.GetResponseStream()
    $reader = New-Object System.IO.StreamReader($result)
    $reader.BaseStream.Position = 0
    $reader.DiscardBufferedData()
    $responseBody = $reader.ReadToEnd()
    Write-Host  $responseBody.errorCode " - " $responseBody.message
    foreach($modelState in $responseBody.modelState)
    {
        $modelState
    }
}
```

REST API Python Scripts

This code sample demonstrates how to use Python 3 to authenticate to Secret Server, retrieve a secret, and update secret fields and items.

This code uses the "requests" module, which makes it simple to communicate via REST. You can install the requests module by running: `pip install requests`

An auth token is required to use this script.

```
import http.client
import urllib
import json
import requests
site = '[Secret Server URL" /> Site]' #ex: http://domain.com/SecretServer
authApi = '/oauth2/token'
api = site + '/api/v1'
token = "<TOKEN>"
# REST call to retrieve a secret by ID
def GetSecret(token, secretId):
    headers = {'Authorization':'Bearer ' + token, 'content-type':'application/json'}
    resp = requests.get(api + '/secrets/' + str(secretId), headers=headers)
    if resp.status_code not in (200, 304):
        raise Exception("Error retrieving Secret. %s %s" % (resp.status_code, resp))
    return resp.json()
# REST call method to update the secret on the server
def UpdateSecret(token, secret):
    headers = {'Authorization':'Bearer ' + token, 'content-type':'application/json'}
    secretId = secret['id']
    resp = requests.put(api + '/secrets/' + str(secretId), json=secret, headers=headers)
    if resp.status_code not in (200, 304):
        raise Exception("Error updating Secret. %s %s" % (resp.status_code, resp))
    return resp.json()
# Retrieves the secret item by its "slug" value
def GetItemBySlug(secretItems, slug):
    for x in secret['items']:
        if x['slug'] == slug:
            return x
    raise Exception('Item not found for slug: %s' % slug)
# Updates the secret item on the secret with the updated secret item
def UpdateSecretItem(secret, updatedItem):
    secretItems = secret['items']
    for x in secretItems:
        if x['itemId'] == updatedItem['itemId']:
            x.update(updatedItem)
            return
    raise Exception('Secret item not found for item id: %s' % str(updatedItem['itemId']))
# Get secret with ID = 1
print("Retrieving Secret with id: 1...")
secret = GetSecret(token, 1)
print("Secret Name: " + secret['name'])
print("Secret ID: " + str(secret['id']))
```

```
print("Active: " + str(secret['active']))
# Get the "Notes" secret item
notesItem = GetItemBySlug(secret, 'notes')
print("Notes secret field value: %s" % notesItem['itemValue'])
print()
# Change value of "Notes" secret item
print("Updating secret...")
notesItem.update({'itemValue': 'New Notes Value'})
UpdateSecretItem(secret, notesItem)
print("Secret updated.")
print()
# Change secret values
updateValues = {'name':'Updated Secret Name' }
secret.update(updateValues)
updatedSecret = UpdateSecret(token, secret)
notesItem = GetItemBySlug(updatedSecret, 'notes')
print("Updated Secret Name: " + updatedSecret['name'])
print("Notes secret field value: %s" % notesItem['itemValue'])
```

REST API Perl Scripts

Secret Server web services can be called using scripts. This example demonstrates how to authenticate and retrieve a Secret programmatically in Perl. If connecting to an Installed instance, change the URL to match your site and specify the application name if configured.

This example runs using standard Perl modules and Strawberry Perl v5.24.1.1.

You may need to install REST::Client from CPAN which can be done by running:

```
cpanm REST::Client
```

This module makes interacting with RESTful APIs much easier.

The TSS script below leverages the API created in the TSS with .pm extension file.

The TokenResponse with .pm extension file defines an object that stores the authentication credentials for the session.

The ErrorResponse with .pm extension file defines an object that stores error information should a REST request fail.

The SecretModel with .pm extension file defines an object that stores the Secret information when requesting a particular Secret from the REST API.

TSS.pl:

```
use REST::Client;
use JSON;
use TSS;
# Enter your <Secret Server URL /> web site here.
my $host = 'http://localhost';;
# Enter the application name, should it be defined.
my $applicationPath = '/secretserver';
```

```
my $false = 0;
my $true = 1;
my $usingSelfSignedCertificate = $true;
if ($usingSelfSignedCertificate) {
    # Disable hostname validation since this will fail with self-singed certs.
    $ENV{'PERL_LWP_SSL_VERIFY_HOSTNAME'} = $false;
}
# Initialize the TSS Perl API with the connection information provided above.
my $tss = TSS->new(
        secretServerUrl => $host,
        applicationPath => $applicationPath
    );
print "Username: ";
my $input = <STDIN>;
chomp $input;
my $username = $input;
print "Password: ";
$input = <STDIN>;
chomp $input;
my $password = $input;
# Authenticate to <Secret Server URL />
my $response = $tss->getToken($username, $password);
if (ref $response eq 'ErrorResponse') {
    $tss->handleError($response);
    exit();
}
# Persist the API token in our TSS API session.
$tss->tokenResponse($response);
$password = undef;
print "Secret ID to retrieve: ";
$input = <STDIN>;
chomp $input;
my $secretId = $input;
# Retrieve a Secret from <Secret Server URL /> by Secret ID.
$response = $tss->getSecret($secretId);
if (ref $response eq 'ErrorResponse') {
    $tss->handleError($response);
    exit();
}
# Read the SecretModel object
print 'Secret ' . $response->id . ' retrieved successfully. Secret name: ' . $response-
>name . "\r\n";
```

TSS.pm:

```
package TSS;
use REST::Client;
use JSON;
use TokenResponse;
use ErrorResponse;
```

```
use SecretModel;
use strict;
sub new {
    my($class, %args) = @_;
    my $self = {
        client => REST::Client->new(),
        secretServerUrl => undef,
        applicationPath => undef,
        tokenResponse => undef,
        %args
    };
    bless $self, $class;
    $self->client->setHost($self->secretServerUrl);
    return($self);
}
sub client {
    my $self = shift;
    return $self->{client};
}
sub secretServerUrl {
    my $self = shift;
    return $self->{secretServerUrl};
}
sub applicationPath {
    my $self = shift;
    return $self->{applicationPath};
}
# Getter and setter so that API tokens can be reused through a session.
sub tokenResponse {
    my $self = shift;
    my $tokenResponse = shift;
    if ($tokenResponse ne undef) {
        $self->{tokenResponse} = $tokenResponse;
        return $tokenResponse;
    }
    return $self->{tokenResponse};
}
# Crafts a HTTP header for providing authentication to <MadCap:variable name="global-
vars.SecretServer" xmlns:MadCap="http://www.madcapsoftware.com/Schemas/MadCap.xsd" />.
sub getHeaders {
    my $self = shift;
    my $headers = {
        Accept => 'application/json',
        Authorization => 'Bearer ' . $self->tokenResponse->access_token
    };
    return $headers;
}
sub handleError {
    my $self = shift;
    my $errorResponse = shift;
    print 'ERROR|' . $errorResponse->message . ' (Error Code: ' . $errorResponse-
>errorCode . ')' . "\r\n";
}
```

```
# Authenticates to <MadCap:variable name="global-vars.SecretServer"
xmlns:MadCap="http://www.madcapsoftware.com/Schemas/MadCap.xsd" /> by POST'ing a Form URL
Encoded content body containing sensitive credentials.
sub getToken {
    my $self = shift;
    my $username = shift;
    my $password = shift;
    my $postParams = $self->client->buildQuery([
            username => $username,
            password => $password,
            grant_type => "password"
        ]);
    my $response = $self->client->POST($self->applicationPath . '/oauth2/token', substr
($postParams, 1), {'Content-Type' => 'application/x-www-form-urlencoded'});
    if ($self->client->responseCode() ne 200) {
        my $errorResponse = ErrorResponse->new(
                errorCode => $self->client->responseCode(),
                message => "Unable to get token."
            );
        return $errorResponse;
    }
    my $responseJson = from_json($response->responseContent);
    if (exists $responseJson->{'errorCode'}) {
        my $errorResponse = ErrorResponse->new(
                errorCode => $responseJson->{'errorCode'},
                message => $responseJson->{'message'}
            );
        return $errorResponse;
    }
    my $tokenResponse = TokenResponse->new(
            access_token => $responseJson->{'access_token'},
            token_type => $responseJson->{'token_type'},
            expires_in => $responseJson->{'expires_in'}
        );
    return $tokenResponse;
}
# Gets a Secret from <Secret Server URL /> by GET'ing a specified Secret by ID and passing
an authentication header containing our API token.
sub getSecret {
    my $self = shift;
    my @params = @_;
    my $secretId = $params[0];
    my $headers = $self->getHeaders();
    $self->client->GET($self->applicationPath . '/api/v1/secrets/' . $secretId, $headers);
    my $responseJson = from_json($self->client->responseContent());
    if (exists $responseJson->{'errorCode'}) {
        my $errorResponse = ErrorResponse->new(
                errorCode => $responseJson->{'errorCode'},
                message => $responseJson->{'message'}
            );
        return $errorResponse;
    }
    my $secretModel = SecretModel->new(
```

```
            id => $responseJson->{'id'},
            name => $responseJson->{'name'},
            secretTemplateId => $responseJson->{'secretTemplateId'},
            folderId => $responseJson->{'folderId'},
            active => $responseJson->{'active'},
            items => $responseJson->{'items'},
            launcherConnectAsSecretId => $responseJson->{'launcherConnectAsSecretId'},
            checkOutMinutesRemaining => $responseJson->{'checkOutMinutesRemaining'},
            checkedOut => $responseJson->{'checkedOut'},
            checkOutUserDisplayName => $responseJson->{'checkOutUserDisplayName'},
            checkOutUserId => $responseJson->{'checkOutUserId'},
            isRestricted => $responseJson->{'isRestricted'},
            isOutOfSync => $responseJson->{'isOutOfSync'},
            outOfSyncReason => $responseJson->{'outOfSyncReason'},
            autoChangeEnabled => $responseJson->{'autoChangeEnabled'},
            autoChangeNextPassword => $responseJson->{'autoChangeNextPassword'},
            requiresApprovalForAccess => $responseJson->{'requiresApprovalForAccess'},
            requiresComment => $responseJson->{'requiresComment'},
            checkOutEnabled => $responseJson->{'checkOutEnabled'},
            checkOutIntervalMinutes => $responseJson->{'checkOutIntervalMinutes'},
            checkOutChangePasswordEnabled => $responseJson->
 {'checkOutChangePasswordEnabled'},
            proxyEnabled => $responseJson->{'proxyEnabled'},
            sessionRecordingEnabled => $responseJson->{'sessionRecordingEnabled'},
            restrictSshCommands => $responseJson->{'restrictSshCommands'},
            allowOwnersUnrestrictedSshCommands => $responseJson->
 {'allowOwnersUnrestrictedSshCommands'},
            isDoubleLock => $responseJson->{'isDoubleLock'},
            doubleLockId => $responseJson->{'doubleLockId'},
            enableInheritPermissions => $responseJson->{'enableInheritPermissions'},
            passwordTypeWebScriptId => $responseJson->{'passwordTypeWebScriptId'},
            siteId => $responseJson->{'siteId'},
            enableInheritSecretPolicy => $responseJson->{'enableInheritSecretPolicy'},
            secretPolicyId => $responseJson->{'secretPolicyId'},
            lastHeartBeatStatus => $responseJson->{'lastHeartBeatStatus'},
            lastHeartBeatCheck => $responseJson->{'lastHeartBeatCheck'},
            failedPasswordChangeAttempts => $responseJson->
 {'failedPasswordChangeAttempts'},
            lastPasswordChangeAttempt => $responseJson->{'lastPasswordChangeAttempt'},
            secretTemplateName => $responseJson->{'secretTemplateName'},
            responseCodes => $responseJson->{'responseCodes'}
        );
    return $secretModel;
 }
 1;
```

TokenResponse.pm:

```
 package TokenResponse;
 use warnings;
```

```
use strict;
sub new {
    my $class = shift;
    my %options = @_;
    my $self = {
        access_token => undef,
        token_type => undef,
        expires_in => undef,
        %options
    };
    bless $self, $class;
    return $self;
}
sub access_token {
    my $self = shift;
    return $self->{access_token};
}
sub token_type {
    my $self = shift;
    return $self->{token_type};
}
sub expires_in {
    my $self = shift;
    return $self->{expires_in};
}
1;
```

ErrorResponse.pm:

```
package ErrorResponse;
use warnings;
use strict;
sub new {
    my $class = shift;
    my %options = @_;
    my $self = {
        errorCode => undef,
        message =>  undef,
        %options
    };
    bless $self, $class;
    return $self;
}
sub errorCode {
    my $self = shift;
    return $self->{errorCode};
}
sub message {
    my $self = shift;
    return $self->{message};
}
```

```
1;
```

SecretModel.pm:

```
package SecretModel;
use warnings;
use strict;
sub new {
    my $class = shift;
    my %options = @_;
    my $self = {
        id => undef,
        name => undef,
        secretTemplateId => undef,
        folderId => undef,
        active => undef,
        items => undef,
        launcherConnectAsSecretId => undef,
        checkOutMinutesRemaining => undef,
        checkedOut => undef,
        checkOutUserDisplayName => undef,
        checkOutUserId => undef,
        isRestricted => undef,
        isOutOfSync => undef,
        outOfSyncReason => undef,
        autoChangeEnabled => undef,
        autoChangeNextPassword => undef,
        requiresApprovalForAccess => undef,
        requiresComment => undef,
        checkOutEnabled => undef,
        checkOutIntervalMinutes => undef,
        checkOutChangePasswordEnabled => undef,
        proxyEnabled => undef,
        sessionRecordingEnabled => undef,
        restrictSshCommands => undef,
        allowOwnersUnrestrictedSshCommands => undef,
        isDoubleLock => undef,
        doubleLockId => undef,
        enableInheritPermissions => undef,
        passwordTypeWebScriptId => undef,
        siteId => undef,
        enableInheritSecretPolicy => undef,
        secretPolicyId => undef,
        lastHeartBeatStatus => undef,
        lastHeartBeatCheck => undef,
        failedPasswordChangeAttempts => undef,
        lastPasswordChangeAttempt => undef,
        secretTemplateName => undef,
        responseCodes => undef,
        %options
```

```
    };
    bless $self, $class;
    return $self;
}
sub id {
  my $self = shift;
  return $self->{id};
}
sub name {
  my $self = shift;
  return $self->{name};
}
sub secretTemplateId {
  my $self = shift;
  return $self->{secretTemplateId};
}
sub folderId {
  my $self = shift;
  return $self->{folderId};
}
sub active {
  my $self = shift;
  return $self->{active};
}
sub items {
  my $self = shift;
  return $self->{items};
}
sub launcherConnectAsSecretId {
  my $self = shift;
  return $self->{launcherConnectAsSecretId};
}
sub checkOutMinutesRemaining {
  my $self = shift;
  return $self->{checkOutMinutesRemaining};
}
sub checkedOut {
  my $self = shift;
  return $self->{checkedOut};
}
sub checkOutUserDisplayName {
  my $self = shift;
  return $self->{checkOutUserDisplayName};
}
sub checkOutUserId {
  my $self = shift;
  return $self->{checkOutUserId};
}
sub isRestricted {
  my $self = shift;
  return $self->{isRestricted};
}
sub isOutOfSync {
```

```
  my $self = shift;
  return $self->{isOutOfSync};
}
sub outOfSyncReason {
  my $self = shift;
  return $self->{outOfSyncReason};
}
sub autoChangeEnabled {
  my $self = shift;
  return $self->{autoChangeEnabled};
}
sub autoChangeNextPassword {
  my $self = shift;
  return $self->{autoChangeNextPassword};
}
sub requiresApprovalForAccess {
  my $self = shift;
  return $self->{requiresApprovalForAccess};
}
sub requiresComment {
  my $self = shift;
  return $self->{requiresComment};
}
sub checkOutEnabled {
  my $self = shift;
  return $self->{checkOutEnabled};
}
sub checkOutIntervalMinutes {
  my $self = shift;
  return $self->{checkOutIntervalMinutes};
}
sub checkOutChangePasswordEnabled {
  my $self = shift;
  return $self->{checkOutChangePasswordEnabled};
}
sub proxyEnabled {
  my $self = shift;
  return $self->{proxyEnabled};
}
sub sessionRecordingEnabled {
  my $self = shift;
  return $self->{sessionRecordingEnabled};
}
sub restrictSshCommands {
  my $self = shift;
  return $self->{restrictSshCommands};
}
sub allowOwnersUnrestrictedSshCommands {
  my $self = shift;
  return $self->{allowOwnersUnrestrictedSshCommands};
}
sub isDoubleLock {
  my $self = shift;
```

```
  return $self->{isDoubleLock};
}
sub doubleLockId {
  my $self = shift;
  return $self->{doubleLockId};
}
sub enableInheritPermissions {
  my $self = shift;
  return $self->{enableInheritPermissions};
}
sub passwordTypeWebScriptId {
  my $self = shift;
  return $self->{passwordTypeWebScriptId};
}
sub siteId {
  my $self = shift;
  return $self->{siteId};
}
sub enableInheritSecretPolicy {
  my $self = shift;
  return $self->{enableInheritSecretPolicy};
}
sub secretPolicyId {
  my $self = shift;
  return $self->{secretPolicyId};
}
sub lastHeartBeatStatus {
  my $self = shift;
  return $self->{lastHeartBeatStatus};
}
sub lastHeartBeatCheck {
  my $self = shift;
  return $self->{lastHeartBeatCheck};
}
sub failedPasswordChangeAttempts {
  my $self = shift;
  return $self->{failedPasswordChangeAttempts};
}
sub lastPasswordChangeAttempt {
  my $self = shift;
  return $self->{lastPasswordChangeAttempt};
}
sub secretTemplateName {
  my $self = shift;
  return $self->{secretTemplateName};
}
sub responseCodes {
  my $self = shift;
  return $self->{responseCodes};
}
1;
```

SDK for DevOps Overview

The SDK for DevOps provides a comprehensive set of tools and resources to facilitate secure and efficient management of secrets within development and operational environments. Below is a brief overview of the key components:

- Using the Secret Server SDK for DevOps

The Secret Server SDK offers a robust framework for integrating secret management into your applications. It provides APIs and libraries that allow developers to interact with Secret Server programmatically, ensuring that sensitive information is handled securely.

- Secret Server SDK Integration

SDK Integration focuses on how to seamlessly incorporate the Secret Server SDK into your existing systems. This section provides guidance on setting up the SDK, configuring it for your environment, and leveraging its capabilities to enhance your security posture.

- SDK for DevOps Downloads

This section provides access to the necessary files and resources required to get started with the Secret Server SDK. It includes download links for SDK packages, documentation, and other essential tools.

- SDK CLI

The SDK Command Line Interface (CLI) offers a powerful way to interact with Secret Server from the command line. It enables automation of secret management tasks, making it easier to integrate with CI/CD pipelines and other automated workflows.

Using the Secret Server SDK for DevOps

Overview

The Secret Server Software Development Kit for DevOps tool, was created for securing and streamlining DevOps processes in regard to Secret Server. The SDK for DevOps tool allows you to efficiently engage Secret Server via a Command Line Interface (CLI) without compromising security. It allows for secure retrieval of credentials from, as well as tracking access to a secure vault.

The SDK uses the "REST API Reference Download" on page 1500. The SDK is a .NET library (available via a NuGet package), which you can use in a custom application. The SDK .NET library exposes a limited subset of the REST API.

> You can download the current and legacy versions of the API on the "Downloads for the Secret Server SDK for DevOps" on page 1551 page.

> The SDK can run on any version of .NET that supports .NET Standard 2.0 apps including: .NET Core 2.0 plus, .NET Framework 4.6.1 plus, .NET Framework 4.8 plus, .NET 5 plus, .NET 6 plus, and .NET 7 plus.

> See the Delinea SDK Integration Doc on GitHub for more information.

The SDK is designed to be used as shown below, not to be run using IWA to retrieve tokens or Secret information. Given this, the SDK is not supported with IWA.

There is also a .NET Core CLI SDK Client that uses the SDK .NET library. The SDK Client was created to allow customers to write automation scripts to access secrets without having to write code to directly access the REST API.

The .NET SDK library and the .NET Core CLI client both:

- Automatically store the credentials and remote server in an encrypted file used to acquire an OAUTH token. The token is then used to make subsequent API calls. OAUTH tokens have an expiration time, which is configurable in the UI on the configuration page via the **Session Timeout for Webservices** value.
- Get the contents of a secret.
- Provide client-side caching (SDK client caching).

Secret Server has user and application accounts. Both types of accounts can access Secret Server via the REST API. Application accounts are not counted for licensing purposes. Application accounts can *only* access Secret Server via the REST API. Both account types never expire. Secret Server provides security for automated clients. SDK rules manage permissions. Client IDs are created when `SecretServerClient.Configure()` or `tss init` is called. The client ID is used to reference SDK client instances.

Do NOT give an application account the Administrator Role or all role permissions. It is recommended to only assign the minimum roles needed to complete the task. That means to create a new Role with only the permissions it needs for the application account and only share the secrets needed.

For REST API Client Generation (Advanced), please see "REST API Client Generation with Swagger" on page 1495.

We have a Python SDK that is independent of the SDK .NET library. It allows a Python script to access secrets without requiring REST knowledge. It has access to a small subset of the REST API.

How it Works

The SDK is a console application written in .NET Core that wires up its own credentials based on the machine it is installed on. Those credentials, called **DevOps Users**, do not have any rights in Secret Server but can be assigned to other Secret Server users or application user accounts, mimicking permissions to access secrets.

This removes the widespread DevOps issues with hard coding credentials into scripts and configuration files. Instead, the target system is registered via IP address which is whitelisted, providing REST authentication without entering user credentials. You can use the SDK to retrieve a REST user token for our REST API, or you can use the SDK to perform direct queries on Secret Server.

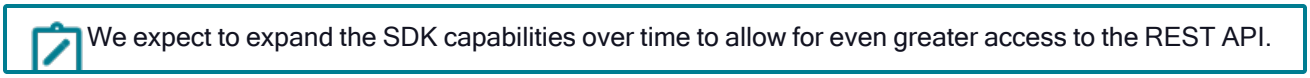
The SDK establishes secure access points so that PowerShell users can employ the Secret Server API directly from the CLI, without wasting time entering privileged account passwords.

Configuration Overview

Secret Server exposes a REST API interface that is used by the SDK client. When the SDK sends a REST request, Secret Server determines where the request came from (via IP address) and what permissions it should be granted (via a rule set in Secret Server). Once the client is initialized with Secret Server and the rule name, this configuration is stored and encrypted on the client machine, ready for subsequent calls.

Out of the box, the SDK offers:

- Token retrieval
- Secret retrieval
- Secret field retrieval



The SDK requires setup in two areas: Secret Server configuration and SDK installation on the DevOps system.

Required Roles and Permissions

- **Administer Configuration**: Allows a user to enable SDK functionality in Secret Server, that is, to enable webservices and the SDK itself.
- **Administer Create Users**: Allows a user to access the **Settings > All settings > Tools and integrations > SDK client** page in Secret Server.
- **Administer scripts**: This permission is required to be able to create an SDK Client Onboarding rule. It allows a user to view and edit PowerShell, SQL, and SSH scripts on the **Scripts Administration** page.
- **Administer Users**: Allows a user to operate the SDK to retrieve account credentials on client machines. Alternatively, you can be the owner of the application account used by the SDK.
- **View Launcher Password**: Allows a user to unmask the password on the view screen of secrets with a launcher. Typically, this includes Web Passwords accounts, Active Directory accounts, Local Windows accounts, and Linux accounts.

Setup Procedure

Task 1: Configuring Secret Server

Configure Secret Server for communication with the SDK

1. Navigate to **Admin > SDK Client**, the SDK Client Management page opens.
2. Under the **Configuration** tab, click **Edit**, and check to select **Enable** SDK Configuration. When done, click **Save**.
3. Navigate to **Admin > Application**.
4. On the Application page click **Edit** and select the **Enable Webservices** check box.
5. Leave the default settings for webservices and click **Save**.

Select or Setup Application Accounts

Select or setup any application accounts that you want for use by SDK clients. Make sure these application accounts have appropriate permissions to access any secrets or execute any operations the client host needs to perform.

To create a new application account with the needed permissions:

1. Go to **Admin > Users**, the User Management page appears.
2. Click the **Create user** button. The **Add User** popup appears.
3. Fill in the following fields:
   a. **Username**: the account name.
   b. **Display name**: the searchable name for the account of your choice, can be the same as the username.
   c. **Domain**: leave as is with the default **Local** value.
   d. **New Password** and **Confirm Password**: a password of your choice.
   e. **Email**: type in an email (optional).
   f. Select the **Application Account** checkbox.
   g. **Multifactor authentication**: leave as is with the default **<None>** value.
   h. Make sure the **Enabled** checkbox is selected.
4. Click **Add User**.
5. Create a new role:
   a. Navigate to **Admin > Roles**.
   b. Click **Create Role**. The **Create Role** popup appears.
   c. Type in a **Name** and make sure the **Enabled** checkbox is selected.
   d. Click **Create Role**. The page for the new role appears.
   e. Assign the **View Secret** permission to this role:
      i. Select the **Permissions** tab and click **Edit**.
      ii. Change the **Scope** selection to **All**, and search for the **View Secret** permission with the search box.
      iii. Select it and click **Save**.
6. Assign the role you just created to the application user created above:
   a. Access the role you just created, it has no users assigned.
   b. Under the Assignment tab click **Edit**, the **Scope** option appears. Change its value to **All**.
   c. Search for the user you previously created, toggle the **Domain** option as needed to show all users.
   d. Add the user by selecting the checkbox by its name and clicking **Save**.
7. Assign the **View Launcher Password** role permission to the application user previously created if this account requires access to secrets with launchers associated to them. Use the same steps for assigning this permission as described for View Secret in the previous steps.

Set up an SDK client rule

1. Navigate to **Admin > SDK client**. This will lead you to the SDK Client Management page.
2. Under the **Client Onboarding** tab, click **Create rule**. The **New Rule** page appears.
3. Type in a relevant name in the **Name** field without any spaces. For example: `ProductionWebApp`.
4. Optionally, leave the **Allowed IP Ranges (CIDR)** field blank, or type in an IPV4 address, or use CIDR notation.

   Secret Server will only allow clients to use this rule if they connect from a valid IP address. If not provided, Secret Server will not enforce IP address restrictions on this rule. We strongly recommend using this feature.

   > There is a 250-character limit, so you can only add a few dozen IP addresses unless you use CIDR notation.

   > IPv6 addresses are not supported.

5. For the **User** drop-down list leave the default **All** value selected.
6. In the search box next to the dropdown, look for the application user you just created and select it.

   > Clients are granted the same permissions as this account within Secret Server. If not provided, an account will be automatically created for clients, but will have no default permissions. You must use an application account (the one you created) for the rule. Application accounts are restricted from logging into the system through the normal user interface and do not count towards your license quota.

7. Select the **Require Onboarding Key** checkbox.

   Clients must provide a generated additional key string when authenticating. If not provided, Secret Server allows any client to use the rule if its IP address is within the specified range. We strongly recommend using this feature.

8. When done, click **Save**.

   The page for the new rule you just created will load automatically.

9. Return to the **SDK Client Management** page, and under the **Client Onboarding** tab click on the rule that you have just created, the rule details will expand to the right of the screen.
10. Select **Show Key**, then copy and save the generated onboarding key value, e.g. `U9Ue9PsMYknfoskQvd/0BYMTaTTx2vtWbB/tHI3exDE=`, for future use.

Task 2: Installing the SDK Client

> This requires Secret Server version 10.4+ or Secret Server Cloud.

> IWA is not supported by the SDK.

> The SDK can run on any version of .NET that supports .NET Standard 2.0 apps including: .NET Core 2.0 plus, .NET Framework 4.6.1 plus, .NET 5 plus, and .NET 6 plus.

1. Access "Downloads for the Secret Server SDK for DevOps" on page 1551 and choose the right kit for your platform. We recommend always using the latest version available.
2. Extract/unzip the SDK zip file you downloaded.
   a. Optionally, to get the SDK NuGet packages, see our documentation on GitHub.
3. For Linux systems, you must make the tss program executable by running `chmod u+x tss`.
4. On Linux systems, you must install libunwind:
   - On Red Hat or Centos, run `sudo yum install libunwind libicu`
   - On Ubuntu, run `sudo apt-get install libunwind-dev`
5. For Windows, open a console window by clicking the Start menu and searching for **cmd**.
6. Inside the Command Prompt type `cd [location of the downloaded zip file]/[location of extraction folder]`.

   E.g. `cd downloads/secretserver-sdk-1.5.9-win-x64`
7. To confirm the SDK client tool is installed and working, run help:
   - On Windows, run `tss --help`
   - On non-Windows systems, run `./tss --help`

Task 3: Connecting to Secret Server

Before the client can retrieve data from Secret Server, it must be initialized to connect to the Secret Server instance. This is a one-time operation on the client machine.

> The SDK connection is per user on the machine. If the user who runs the SDK is different than the user who initialized it, the SDK will not work and will need to be re-registered.

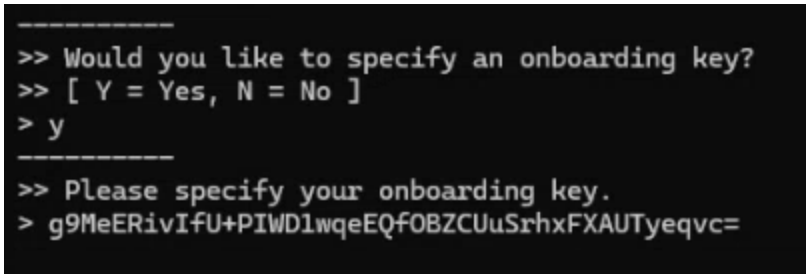
To initialize your connection:

1. Click the Start menu and search for **cmd**.
2. Inside the Command Prompt type `cd [location of the downloaded zip file]\[location of extraction folder]`.

   E.g. `cd downloads\secretserver-sdk-1.5.9-win-x64`
3. Run `tss -i` to use interactive mode:

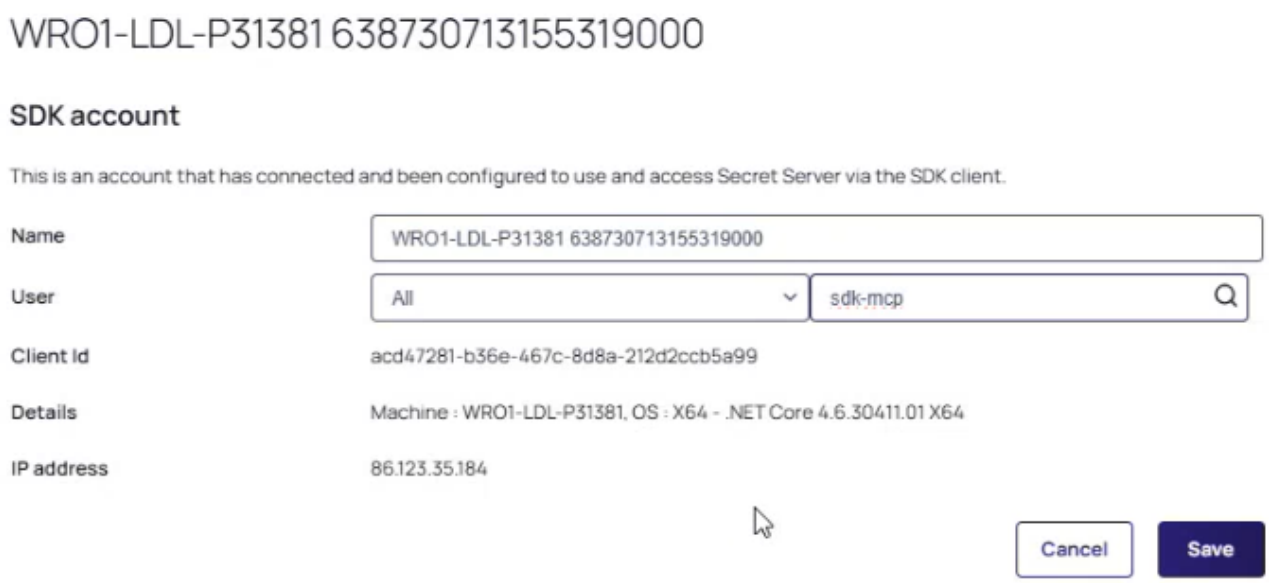
4. Type `init` and press enter.
5. When prompted for the URL, copy and paste the URL of your Secret Server instance, and press enter.

   E.g. https://[secret server url].
6. You are then prompted to specify the rule you created previously in the SDK Client Management page. Type in the rule name and press enter. E.g `ProductionWebApp`.
7. You are now asked if you wish to specify an onboarding key. Enter **yes**.
8. The prompt to specify the key appears. The key was copied after you created the rule. E.g.: `CNrQwRBscnq4qAZ6v3EIAcE27vQuLlz6KSpfRJHryyA=`. You need to paste it in the Command Prompt window and press enter:



9. The "Your SDK Client account registration is complete" message appears. Keep this window open, do not exit.
10. Return to your Secret Server instance and to the SDK Client Management page.
11. In the **Accounts** tab, click on the account that now appears after performing SDK initialization. The side menu appears with information about the account.
12. Click **View details**.
13. Inside the SDK account page, click **Edit**.

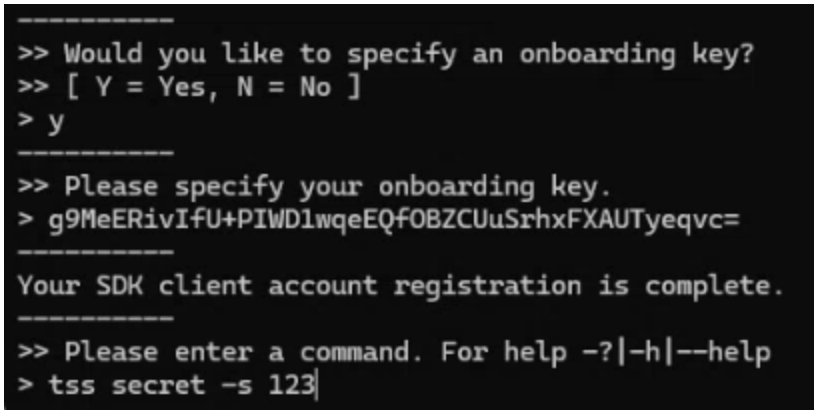
14. Search for the application account/user you previously created in Task 1, e.g:



15. Add the application account username in the box for the **User** search field, it will return a list of users. Pick the one you need and click **Save**.

> If this is not done the SDK Client will return this error: `invalid_client`.

16. Return to the Command Prompt window you kept open. You will now enter the command to view your secret ID `tss secret -s [secret id]`:



17. The prompt to specify a field appears, enter **no**.
18. The Secret will successfully display in the tss application.

Secret Server verifies the client's credentials and IP address restriction (if specified). Once validated, the client and server establish a connection allowing the client to retrieve data from Secret Server.

> **Note:** To perform this operation outside of interactive mode, run `tss init --url [url] -r [rule] -k [key]` using the parameters you recorded earlier for your instance.
>
> Example of operation and parameters to run for connection initialization:
>
> ```
> tss init --url https://myserver/SecretServer/ -r ProductionWebApp -k
> CNrQwRBscnq4qAZ6v3EIAcE27vQuLlz6KSpfRJHryyA=
> ```

Usage Examples

- Retrieving a secret by ID (returns a JSON structure describing the entire secret record): `tss secret -s 4`
- Retrieving all secret field values for a secret by ID: `tss secret -s 4 -ad`
- Retrieving only the value of a particular secret field by secret ID: `tss secret -s 4 -f password`
- Writing a secret field value to a file: `tss secret -s 4 -f password -o passwordfile.txt`
- Retrieving an access token for use in other REST API requests: `tss token`
- Retrieving a secret by ID when that Secret requires a comment: `tss secret -s 4 -c comment`

The SDK client includes an interactive mode (`tss -i`) that allows you to input multiple commands into a series of prompts. To exit interactive mode, run the `exit` command.

SDK Client Management

To view and manage a list of connected SDK clients from within Secret Server:

1. Navigate to **Admin > SDK client**. The SDK Client Management page appears.
2. In the **Accounts** tab a list of connected SDK clients appears for all users if available. You can remove or rename them. You can also filter clients by User.
3. Select the **Client Onboarding** tab to create and manage client onboarding rules.
4. Select the **Configuration** tab to disable or enable all SDK client activity.
5. Select the **Audit** tab to see date and time-stamped SDK client activity concerning onboarding rules and client secrets.
6. To remove the Secret Server connection from a client machine, run the `tss remove` command. This deletes the connection, and the client can no longer retrieve Secret Server data without being re-initialized.

> If you remove a connected SDK client, the user may still be able to reconnect using the client onboarding rule key unless you alter or remove the rule that the client used from Secret Server.

> When an SDK client connects to Secret Server, a client ID is generated. These credentials can be revoked, but will remain valid for 15 minutes after revocation, after which their validity expires.

SDK Client Caching

Overview

To increase performance and reliability, you can configure the SDK client to cache values retrieved from Secret Server on the client machine. Cached values are stored encrypted on disk. You can configure client caching in one of four ways:

- **Never** (0): With this default setting, the client never caches Secret Server data. All data requests result in a query directly against the Secret Server instance. If the instance is unavailable, the requests fail.
- **Server Then Cache** (1): With this setting, the client first attempts to retrieve the value from the server. If the server is unavailable, it checks to see if the same value has been previously fetched within a given period, and if so, it will return the cached value. Use this setting to guard against losing connection to Secret Server.
- **Cache Then Server** (2): With this setting, the client first checks to see if the same value has been previously fetched within a given period. If so, it returns the value without consulting the server. If not, it fetches, caches, and returns the value from the server. Use this setting for increased performance by reducing requests sent to Secret Server.
- **Cache Then Server Fallback on Expired Cache** (3): This strategy operates similarly to "Cache Then Server," but if the server is unavailable and an expired value exists in the cache, the client returns that value as a last resort. Use this strategy for increased performance and reliability.

All these cache strategies have a configurable age, in minutes, after which the value is considered expired and not used (except in **Cache Then Server Fallback** mode). Cache settings are set using the client application, see the examples below for more details.

Examples

- Turn off caching: `tss cache --strategy 0`
- Turn on the **Cache Then Server** setting with a cache age of five minutes: `tss cache --strategy 2 --age 5`
- Immediately clear all cached values: `tss cache --bust`
- Show the current cache settings: `tss cache --current`

> Anytime you use a cached value, recent changes made to Secret Server may not be applied, including changes to the value itself, permissions, or other access control settings. Examine your organization's security policies and application requirements to determine the best cache settings to use.

Secret Server SDK Integration

SDK Integration in a C# Project

> In this scenario we assume we can recompile the application, and will dynamically retrieve passwords from the vault whenever needed.

Prerequisites

1. Create a rule in Secret Server for client onboarding:
   a. Navigate to **Admin > SDK Client**, the SDK Client Management page opens.
   b. Under the **Client Onboarding** tab, click **Create Rule** to create a new rule.
   c. Name your rule (something that helps identify it, such as the application name).
   d. Assign IPv4 restrictions (optional).
   e. Assign an application user account (create one if you have not already).
   f. Generate a Rule Key (optional).
   g. Click to select the **Require Onboarding Key** check box (optional).
   h. Click the **Save** button.
2. The application account you created needs to have access to the secrets your application needs. Ensure the permissions are accurate at the folder or secret level.
3. Download the SDK packages from NuGet. You can do this either directly from Visual Studio (recommended) or manually download and install them from Nuget.org. The latest version is almost always preferred.
   On the Nuget site:
   a. Search for "Thycotic" (an older name for one of the companies that became Delinea)
   b. Install the following packages:
      - Thycotic.SecretServer.SDK
      - Thycotic.SecretServer.SDK.Extension.Configuration
      - Thycotic.SecretServer.SDK.Extension.Integration

> The SDK Nuget packages target both .NET Framework v4.5 or higher, or .NET Standard 2.0. Visual Studio will install the appropriate version based on your project type. .NET Standard 2.0 supports either:
>
> - .NET Core 2.0 if building a .NET Core application
> - .NET (Full Framework) 4.6.1

4. In your project, add the following references:

```
using Thycotic.SecretServer.Sdk.Extensions.Integration.Clients;
using Thycotic.SecretServer.Sdk.Extensions.Integration.Models;
using Thycotic.SecretServer.Sdk.Infrastructure.Models;
```

   Alternatively, you can instantiate the objects and let Visual Studio add the references for you.

Configuring the SDK

You need to instantiate a new SecretServerClient object to interact with the SDK. Below is sample code:

```
var client = new SecretServerClient();
//configure if not configured
client.Configure(new ConfigSettings
  {
    SecretServerUrl = string,
    RuleName = string,
    RuleKey = string,
    CacheStrategy= CacheStrategy.enum,
    CacheAge= int,
    ResetToken= string
  });
```

The code above registers the integrated client with Secret Server. Below is an explanation of the properties of the client's ConfigSettings:

- `ConfigSettings` is an object
- `SecretServerUrl` is the base URL for your Secret Server (if you have a load balancer then it should be the load balanced URL)
- `RuleName` is the name of the rule we created earlier
- `RuleKey` is the key we generated, can be NULL if no key
- `CacheStrategy` is how the SDK will cache requests. This is an enum with four options:
  - `CacheThenServer`
  - `CacheThenServerAllowExpired` (This allows fallback in case Secret Server is not available and the cache is expired, the SDK will still use the cache until it can contact Secret Server and prime the cache)
  - `Never`
  - `ServerThenCache` (This mode is for redundancy since it will fall back to cache in case Secret Server is not available)
- `CacheAge` is how long the cache is valid before it expires in minutes
- `ResetToken` is a random string to revoke clients and reinitialize them

Once the client is configured for the first time, a series of encrypted configuration files are created. By default they will be saved in the current working directory of your application. This path can be customized with the SecretServerSdkConfigDirectory AppSetting.

If using the .NET Standard version of the SDK, these config files are protected by an encryption key. It is saved in the current user's home directory by default, but this path can also be customized with the SecretServerSdkKeyDirectory AppSetting if necessary.

If you need to change your configuration, change your reset token and the SDK will reinitialize the config files. Otherwise, the SDK will not reconfigure itself as long as it can still access and decrypt the config files. Calls to client.Configure() are ignored if the reset token remains the same and the client has already been configured.

Using the SDK

Now we can use the client to get a secret, token, or a secret field from Secret Server, and pass that to our application:

```
var password = client.GetSecretField(<SECRET ID>,"password"); //replace <SECRET ID>
```

The code above retrieves a password from Secret Server, which we can then pass to a connection string or anywhere a password is needed.

You can call the `GetSecret();` method on the client object to get the full secret object, and then access the items property which holds a collection of the secret fields and their values. How you access these values is up to you, but you can use Linq to query what you are looking for. For instance:

```
var secret = client.GetSecret(<SECRET ID>); //replace <SECRET ID>
var server = secret.Items.First(x => x.Slug == "server").ItemValue;
var username = secret.Items.First(x => x.Slug == "username").ItemValue;
var password = secret.Items.First(x => x.Slug == "password").ItemValue;
var database = secret.Items.First(x => x.Slug == "database").ItemValue;
```

and then use these variables to build a connection string:

```
SqlConnectionStringBuilder builder = new SqlConnectionStringBuilder(GetConnectionString
())
{
    ConnectionString = $"server={server};user id={username};password={password};initial
catalog={database}"
};
```

Here is the complete code sample:

```
using System;
using System.Linq;
using Thycotic.SecretServer.Sdk.Extensions.Integration.Clients;
using Thycotic.SecretServer.Sdk.Extensions.Integration.Models;
using Thycotic.SecretServer.Sdk.Infrastructure.Models;
using System.Data.SqlClient;
namespace SDK.Integration
{
    class Program
    {
        static void Main(string[] args)
        {
            var url = "";       //replace with Secret Server URL
            var ruleName = "";  //replace with SDK rule name
            var ruleKey = "";   //replace with on boarding key
            var cacheAge = int; //replace with number for cache age
            var secretId = int; //replace with Secret ID
            var client = new SecretServerClient();
            //configure if not configured
            client.Configure(new ConfigSettings
```

```
            {
                SecretServerUrl = url,
                RuleName = ruleName,
                RuleKey = ruleKey,
                CacheStrategy= CacheStrategy.CacheThenServerAllowExpired,
                CacheAge= cacheAge, //in minutes
                ResetToken= "Token"
            });
            var secret = client.GetSecret(secretId);
            var server = secret.Items.First(x => x.Slug == "server").ItemValue;
            var username = secret.Items.First(x => x.Slug == "username").ItemValue;
            var password = secret.Items.First(x => x.Slug == "password").ItemValue;
            var database = secret.Items.First(x => x.Slug == "database").ItemValue;
            //connection string example
            SqlConnectionStringBuilder builder = new SqlConnectionStringBuilder
(GetConnectionString())
            {
                ConnectionString = $"server={server};user id={username};password=
{password};initial catalog={database}"
            };
            Console.WriteLine(builder.ConnectionString);
            Console.ReadKey();
        }
        private static string GetConnectionString()
        {
            return "Server=(local);Integrated Security=SSPI;" +
                "Initial Catalog=AdventureWorks";
        }
    }
}
```

SDK Integration in web.config

In this scenario, we assume we cannot recompile the application or prefer not to. The use case is as follows:

ASP.NET web application and is a .NET Standard 2.0 application

- We have a ConnectionString(s) inside of our config file that contains plaintext passwords
- We have appSettings with plaintext passwords that our app uses to connect to external services

The SDK allows us to pull data from Secret Server and inject it in the config file.

> The injected data is not available in Application_Start, because this is a HTTP Intercept module it will not have ran yet.

Prerequisites for web.config

1. Create a rule in Secret Server for client onboarding using the same method detailed earlier.
2. Install the Thycotic.SecretServer.Sdk.Extensions.Integration.HttpModule Nuget package:
   - This is available in the Nuget package manager from Thycotic. Installing it will also install the necessary dependencies.
   - We recommend this method of installation so dependencies will also be automatically installed, and the Nuget package manager will show when updates are available.
3. It is also possible to manually download the packages from Nuget, but this requires more effort and you won't be notified of updates.
4. The following packages are needed:
   - Thycotic.SecretServer.Sdk
   - Thycotic.SecretServer.Sdk.Extensions.Configuration
   - Thycotic.SecretServer.Sdk.Extensions.Integration
   - Thycotic.SecretServer.Sdk.Extensions.Integration.HttpModule
5. After downloading, use 7-Zip or the like to unpack the ngpkg and navigate to the lib directory
6. Make sure you only extract dlls from subdirectories in NetStandard2.0 or net461
7. Copy the extracted dlls to your application's bin folder:
   - Thycotic.SecretServer.Sdk.dll
   - Thycotic.SecretServer.Sdk.Extensions.Configuration.dll
   - Thycotic.SecretServer.Sdk.Extensions.Integration.dll
   - Thycotic.SecretServer.Sdk.Extensions.Integration.HttpModule.dll

Configuring the SDK for web.config

1. Open your web.config file in your preferred code editor and add the inside your appSettings tag following:

```
<appSettings>
    <add key="tss:CacheAge" value="<cache-age>" />
    <add key="tss:CacheStrategy" value="<cache-strategy>" />
    <add key="tss:SecretServerUrl" value="<your-secret-server-url>" />
    <add key="tss:RuleName" value="<rule-name>" />
    <add key="tss:RuleKey" value="<rule-key>" />
    <add key="tss:ResetToken" value="<reset-token>" />
</appSettings>
```

   - This will configure the SDK to talk to Secret Server, attach it to a rule, authenticate with the optional pre-shared key, configure caching, and add a reset token for reinitialization. Below is an explanation of these

key-value pairs:

- tss:CacheAge: cache age in minutes, that is how long should the SDK keep the cache before trying to refresh
- tss:CacheStrategy: should the SDK cache or not? Strategies are numbered 0 - 3
  - 0: Never cache
  - 1: Server then cache
  - 2: Cache then server
  - 3: Cache then server, but allow expired cache if the server is unreachable
- SecretServerUrl: Your Secret Server URL
- tss:RuleName: the name of the rule you created in Secret Server
- tss:RuleKey: the pre-shared key you generated for the rule
- tss:ResetToken: This is a string value used to reinitialize the SDK. It can be anything, and changing it will cause the client to reinitialize and reregister itself

2. Scroll to:

```
<system.webServer>
    <module>
```

and add the following:

```
<remove name="ThycoticInterceptModule" />
<add name="ThycoticInterceptModule"
type=
"Thycotic.SecretServer.Sdk.Extensions.Integration.HttpModule.Modules.ThycoticInterceptM
odule,Thycotic.SecretServer.Sdk.Extensions.Integration.HttpModule" />
```

3. Our section should look like this:

```
<system.webServer>
    <validation validateIntegratedModeConfiguration="false" />
    <modules>
      <remove name="ThycoticInterceptModule" />
      <add name="ThycoticInterceptModule"
          type=
"Thycotic.SecretServer.Sdk.Extensions.Integration.HttpModule.Modules.ThycoticInterceptM
odule,Thycotic.SecretServer.Sdk.Extensions.Integration.HttpModule" />
      <remove name="TelemetryCorrelationHttpModule" />
      <add name="TelemetryCorrelationHttpModule"
          type="Microsoft.AspNet.TelemetryCorrelation.TelemetryCorrelationHttpModule,
Microsoft.AspNet.TelemetryCorrelation"
          preCondition="integratedMode,managedHandler" />
      <remove name="ApplicationInsightsWebTracking" />
      <add name="ApplicationInsightsWebTracking"
```

```
            type="Microsoft.ApplicationInsights.Web.ApplicationInsightsHttpModule,
    Microsoft.AI.Web"
            preCondition="managedHandler" />
        </modules>
      </system.webServer>
```

Important: You may need to add a binding redirect if you run into:

```
Message: System.IO.FileLoadException : Could not load file or assembly
'System.Runtime.InteropServices.RuntimeInformation, Version=0.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a' or one of its dependencies. The located assembly's manifest
definition does not match the assembly reference. (Exception from HRESULT: 0x80131040)
```

Simply add the following to your `web.config` file:

```
<dependentAssembly>
    <assemblyIdentity
name="System.Runtime.InteropServices.RuntimeInformation"    publicKeyToken=
"b03f5f7f11d50a3a" culture="neutral" />
    <bindingRedirect oldVersion="0.0.0.0-4.0.2.0" newVersion="4.0.2.0" />
</dependentAssembly>
```

4. Match the version number with what you have installed on your system.

Using web.config

To replace plaintext credentials in your `web.config` file we simply use string interpolation to replace the hard-coded values with Secret Server values as shown below.

Example old ConnectionString:

```
<connectionStrings>
    <clear />
    <add
        name="AdventureWorks2014ConnectionString"
        connectionString="Data Source=sql01.domain.com;Initial
Catalog=AdventureWorks2014;Persist Security Info=True;User ID=sa;Password=SgW#5)zLpo($@d"
        providerName="System.Data.SqlClient"
    />
  </connectionStrings>
```

Example new ConnectionString with the Secret Server SDK:

```
<connectionStrings>
    <clear />
    <add
        name="AdventureWorks2014ConnectionString"
        connectionString="Data Source=${server};Initial Catalog=${database};Persist
Security Info=True;User ID=${username};Password=${password}?3112"
        providerName="System.Data.SqlClient"
```

```
  />
</connectionStrings>
```

The ?3112 is the secret ID preceded by a question mark.

SDK API: The SecretServerClient() Class

This class has the following methods:

.BustCache()

This method doesn't have an overload. Calling it destroys the SDK's cache of secrets. The SDK configuration is still retained.

.Configure(IConfigSettings settings, [bool force = false])

```
settings
Type: Object
Key value pairs to configure the SDK


force (optional)
Type: boolean
Default: false
Forces the SDK to reconfigure itself
```


```
force (optional)
Type: boolean
Default: false
Forces the SDK to reconfigure itself
```


.GetSecret(int id)

This method returns a secret object based on the REST secret model.

.GetSecretField(int id, string fieldslug)

This method gets a specific field from the secret instead of returning the whole object.

```
id
Type: int32
The secret id needed to retrieve the secret


fieldslug
Type: String
Slug identifier for the secret field password
```

.GetAccessToken()

This method returns a REST API token from Secret Server, after authenticating with the configured SDK account.

.GetAccessTokenAsync()

Just like GetAccessToken but asynchronous.

SDK Client

To get the command-line interface SDK Client tool check out "Downloads for the Secret Server SDK for DevOps" below. This tool is not required to use the SDK NuGet packages.

The SDK Client and SDK NuGet packages are two separate projects with separate versioning. We actively work to maintain feature parity between the two projects, but at times their features may differ.

Downloads for the Secret Server SDK for DevOps

Overview

The Secret Server SDK replaces and improves upon the existing functionality of the .NET/Application API. Users can leverage this SDK to tokenize credentials in scripts and configuration files for .NET web applications. The SDK can also call for a REST Webservices authentication token for added functionality. Finally, the SDK has a local encrypted cache for every location it is installed on to allow for quicker transit times and resiliency in case communication with Secret Server is lost.

To download the SDK command line tool, choose a version and OS platform below. See the "Using the Secret Server SDK for DevOps" on page 1533 topic for how to use the SDK.

> Due to Java's security flaws and the existence of the Secret Server SDK, we will no longer support the Secret Server Java API. If you use the Secret Server Java API, please consider transitioning to the Secret Server SDK.

SDK Client version 1.5.9

Release Notes

Improved error handling after an Invalid Client error occurs.

Downloads

- Windows x64
- MacOS x64
- Linux x64 (including RHEL 7 to 9)
- Red Hat Enterprise Linux 6 x64

Download File Hashes

Windows x64

- File: `secretserver-sdk-1.5.9-win-x64.zip`
- SHA256: 79C9EF7BA47CB2BACC5902545F10BF79D777D28C56538BB68D7B2074F395B8DF

MacOS x64

- File: `secretserver-sdk-1.5.9-osx-x64.zip`
- SHA256: C27D14073006649B229FFDDDD78A5E86218E9CF17CD8B24F326ABFB9F93185CF

Linux x64

- File: `secretserver-sdk-1.5.9-linux-x64.zip`
- SHA256: C257AF98D1792AF9CF208EDA5BEDA5C6DA38CAE396552F92DF5E4DA4D80EAA36

Red Hat Enterprise Linux 6 x64

- File: `secretserver-sdk-1.5.9-rhel.6-x64.zip`
- SHA256: 8255E2B0A9ED894C4056D82D0507002D41583B502835782578461409F4ED6A36

Legacy Releases

SDK Client version 1.5.7

Release Notes

Added capability to enter a comment when accessing a Secret.

Downloads

- Windows x64
- MacOS x64
- Linux x64 (including RHEL 7 to 9)
- Red Hat Enterprise Linux 6 x64

Download File Hashes

Windows x64

- File: `secretserver-sdk-1.5.7-win-x64.zip`
- SHA256: A932CD4B0CC343AEC83BEF43396F6D5B9F973510963ECC4B22426D5878E84E4C

MacOS x64

- File: `secretserver-sdk-1.5.7-osx-x64.zip`
- SHA256: BCBDA0F0AF86D55DC7EBBCD8BF2B67049D24D58D0AFA2A17EABF0972AFA55A4

Linux x64

- File: `secretserver-sdk-1.5.7-linux-x64.zip`
- SHA256: C5EE5A15CABA1C23C44C65DC4225C282A4D927ECAF0E9E1AEE1BC1ABE7D8BE4E

Red Hat Enterprise Linux 6 x64

- File: `secretserver-sdk-1.5.7-rhel.6-x64.zip`
- SHA256: C3317EEE032EBCF80566A0F66A74A6591EC3689B260B03E55B34B3CDC7DA0E0B

SDK Client version 1.5.5

Release Notes

- Reset the token expiration calculation to use UTC time to avoid unexpected expiration.
- Reworked error handling for secret field-value requests so that the correct error string is extracted from the result if the secret is not accessible.
- Fixed the issue that caused the "tss version" command to return an inaccurate value.
- Updated references to external libraries within the SDK to address reported vulnerabilities.

Downloads

- Windows x64
- MacOS x64
- Linux x64 (including RHEL 7 to 9)
- Red Hat Enterprise Linux 6 x64

Download File Hashes

Windows x64

- File: `secretserver-sdk-1.5.5-win-x64.zip`
- SHA256: FBE984292F5F228630B62E72E2F26A5D66B1C53964BC0707E718D8937DC538DD

MacOS x64

- File: `secretserver-sdk-1.5.5-osx-x64.zip`
- SHA256: A9A7DEEC905D3C21C95BB2A8467FE93640F491E7643760622C5F33B1B5C717C0

Linux x64

- File: `secretserver-sdk-1.5.5-linux-x64.zip`
- SHA256: 4C2C293D9F3F3BE08722FEAF10BD1506D67F2258DD8DAF51FC66D7D13A58AF0E

Red Hat Enterprise Linux 6 x64

- File: `secretserver-sdk-1.5.5-rhel.6-x64.zip`
- SHA256: C8B140D0F746D48EF1938D2836B5C69D1E9A5FAD3817BAF11FF48011F8E2299C

SDK Client version 1.5.4

Release Notes

Added compatibility with OpenSSL 3.0 on Red Hat Linux (RHEL) 9. Resolved the vulnerability relating to System.Text.Encodings.Web.

Downloads

- Windows x64
- MacOS x64
- Linux x64 (including RHEL 7 to 9)
- Red Hat Enterprise Linux 6 x64

Download File Hashes

Windows x64

- File: `secretserver-sdk-1.5.4-win-x64.zip`
- SHA256: FA61DCCC629312397E16A87CCFE355C3DCCF6149E29C36F609A4B383E31AF777

MacOS x64

- File: `secretserver-sdk-1.5.4-osx-x64.zip`
- SHA256: 8D570AB83FE2E3C33AAF390B8082E270FBCFBCD865648BCCB154832FA2707DC8

Linux x64

- File: `secretserver-sdk-1.5.4-linux-x64.zip`
- SHA256: F4A2C0553597BF763E7F33FB95EE5A507BF7CB4187FBA8C8CDF4BEE5AFB6A406

Red Hat Enterprise Linux 6 x64

- File: `secretserver-sdk-1.5.4-rhel.6-x64.zip`
- SHA256: 1D4C5CC664458F45FE7010D531910B7B3373FF4B02A6B06C9507B5818CECDE81

SDK Client version 1.5.3

Release Notes

Added a "multi" option to the CLI. You can now select one field from multiple secrets in a single call by passing a comma-separated list of multiple Secret IDs and a field name.

Downloads

- Windows x64
- MacOS x64
- Linux x64 (including RHEL 7 to 8–RHEL 9 uses OpenSSL v3, which the SDK does not support)
- Red Hat Enterprise Linux 6 x64

Download File Hashes

Windows x64

- File: `secretserver-sdk-1.5.3-win-x64.zip`
- SHA256: 231F88106CC03434AC7E107579BB1B7686B10011F208068913F5F3762A4A5B64

MacOS x64

- File: `secretserver-sdk-1.5.3-osx-x64.zip`
- SHA256: E13657A9E80235F2404F398254E537185619DCF921546C03D28BDF17C75E1694

Linux x64

- File: `secretserver-sdk-1.5.3-linux-x64.zip`
- SHA256: 8AFAB562C35EDC4AFD5EB6B6275635A06C319FD0275318B5DE6ED61D8678FCC4

Red Hat Enterprise Linux 6 x64

- File: `secretserver-sdk-1.5.3-rhel.6-x64.zip`
- SHA256: E4E83E632FFA640A47C70A880E53DBDE38F12F2CAF6FE4441D18ECEB7D716904

SDK Client version 1.5.0

Release Notes

- Cache improvements
- Stability improvements

Downloads

- Windows x64
- MacOS x64

- Linux x64 (including RHEL 7 to 8–RHEL 9 uses OpenSSL v3, which the SDK does not support)
- Red Hat Enterprise Linux 6 x64

Download File Hashes

Windows x64

- File: `secretserver-sdk-1.5.0-win-x64.zip`
- SHA256: 4B3246470E4EA87190CE3B511151E93E04E7363EADBD145BA0EE8AA63BC1378B

MacOS x64

- File: `secretserver-sdk-1.5.0-osx-x64.zip`
- SHA256: 492D177CF86554EC22B947957328D833471E2DECFA7AE95FC0D2B3FFB1B24E37

Linux x64

- File: `secretserver-sdk-1.5.0-linux-x64.zip`
- SHA256: 94672CA26C438301A070C020FFAEE2932250F3358C91B30BC4F9B8F0E0A1210B

Red Hat Enterprise Linux 6 x64

- File: `secretserver-sdk-1.5.0-rhel.6-x64.zip`
- SHA256: BF0BA52C7BA0838E56C4C0C44FC29794223B04336458D342F14872FE3914E30B

SDK Client version 1.4.1

Release Notes

- New option to specify a configuration and cache directory other than the default.
- SDK CLI (`tss`) now has a `--configure-directory` option.
- NuGet SDK packages now check for a `SecretServerSdkKeyDirectory AppSetting`.
- Updated the NuGet SDK packages to support customizing the key storage directory. They now check for a `SecretServerSdkKeyDirectory AppSetting`.
- Updated the NuGet SDK packages to have a `SecretServerClientGetAccessToken()` and `GetAccessTokenAsync()` method, to get a REST or SOAP API token. This is equivalent to the existing SDK CLI command `token`.

Downloads

- Windows x64
- MacOS x64
- Linux x64 (including RHEL 7 to 8–RHEL 9 uses OpenSSL v3, which the SDK does not support)
- Red Hat Enterprise Linux 6 x64

SDK Client version 1.3.0

Release Notes

- Upgrade to .NET Core 2.1 Runtime.
- Encrypt configuration with DPAPI on Windows by default.
- New commands to display connection status and version.
- New option to specify a key storage directory other than the default.
- Target multiple OS versions with a single build.
- New build for Red Hat Enterprise Linux 6.

Downloads

- Windows x64
- MacOS x64
- Linux x64 (including RHEL 7 to 8–RHEL 9 uses OpenSSL v3, which the SDK does not support)
- Red Hat Enterprise Linux 6 x64

SDK Client version 1.0.0

Release Notes

Initial release of the SDK command line tool.

Downloads

- Windows 10 x64
- CentOS 7 x64
- Red Hat Enterprise Linux 7 x64
- Ubuntu 16.10 x64
- MacOS 10.12 x64

SDK NuGet Packages (Optional)

To get the SDK NuGet packages, check out the documentation on our GitHub page. This package is not required to use the SDK Client.

> The SDK Client and SDK NuGet packages are two separate projects with separate versioning. We actively work to maintain feature parity between the two projects, but at times their features may differ.

Secret Server CLI Client Reference

The TSS CLI client is an integration utility that allows you to interact with Secret Server. This guide provides a quick reference for the available commands and options.

The client is abbreviated as TSS because originally is was the *Thycotic Secret Server* client. Thycotic was one of the companies that became Delinea.

Basic Usage

```
tss [options] [command]
```

Global Options

Switch	Abbreviated Switch	Purpose
--config-directory	-cd	Set the storage directory for the config files
--help	-h, -?	Show help information
--interactive	-i	Enable interactive mode
--key-directory	-kd	Set the storage directory for the config file encryption key
--verbose	-v	Output verbose errors

Available Commands

cache

Manage the cache strategy for this instance.

```
tss cache [options]
```

Switch	Abbreviated Switch	Purpose
--age	-a	The cache age defines how long an item can live
--bust	-b	Bust the local cache
--config-directory	-cd	Set the storage directory for the config files
--current	-c	Get the current cache settings
--help	-h, -?	Show help information
--interactive	-i	Enable interactive mode

Switch	Abbreviated Switch	Purpose
--key-directory	-kd	Set the storage directory for the config file encryption key
--strategy	-s	The cache strategy to use (Never = 0, Server then cache = 1, Cache then server = 2, Cache then server allow fallback on expired cache = 3)
--verbose	-v	Output verbose errors

exit

Exit the Secret Server CLI.

```
tss exit [options]
```

Switch	Abbreviated Switch	Purpose
--config-directory	-cd	Set the storage directory for the config files
--help	-h, -?	Show help information
--interactive	-i	Enable interactive mode
--key-directory	-kd	Set the storage directory for the config file encryption key
--verbose	-v	Output verbose errors

init

Initialize this machine to communicate with your Secret Server.

```
tss init [options]
```

Switch	Abbreviated Switch	Purpose
--config-directory	-cd	Set the storage directory for the config files
--help	-h, -?	Show help information
--if-not-exist	-e	Do not generate an error if already initialized
--interactive	-i	Enable interactive mode

Switch	Abbreviated Switch	Purpose
--key-directory	-kd	Set the storage directory for the config file encryption key
--onboarding-key	-k	The onboarding key for the rule
--rule-name	-r	The name of the rule that should be matched
--url	-u	The Secret Server URL (https://<name>)
--verbose	-v	Output verbose errors

multi

Get the value of a field from one or more specified secrets.

```
tss multi [options]
```

Switch	Abbreviated Switch	Purpose
--as-dictionary	-ad	Format secret as a dictionary of secret field/value pairs
--config-directory	-cd	Set the storage directory for the config files
--field	-f	The secret field's slug
--help	-h, -?	Show help information
--interactive	-i	Enable interactive mode
--key-directory	-kd	Set the storage directory for the config file encryption key
--output	-o	The optional output location
--secrets	-s	Comma-separated list of secret IDs (id1,id2,id3,...)
--verbose	-v	Output verbose errors

remove

Remove configuration settings.

```
tss remove [options]
```

Switch	Abbreviated Switch	Purpose
--config-directory	-cd	Set the storage directory for the config files
--confirm	-c	Automatically confirm this action without a confirmation prompt
--help	-h, -?	Show help information
--interactive	-i	Enable interactive mode
--key-directory	-kd	Set the storage directory for the config file encryption key
--verbose	-v	Output verbose errors

secret

Get the value of a field from the specified secret.

```
tss secret [options]
```

Switch	Abbreviated Switch	Purpose
--as-dictionary	-ad	Format secret as a dictionary of secret field/value pairs
--comment	-c	Add a comment
--config-directory	-cd	Set the storage directory for the config files
--field	-f	The secret field's slug
--help	-h, -?	Show help information
--interactive	-i	Enable interactive mode
--key-directory	-kd	Set the storage directory for the config file encryption key
--output	-o	The optional output location
--secret	-s	The ID of the secret
--verbose	-v	Output verbose errors

status

Display the current connection status of the SDK client.

```
tss status [options]
```

Switch	Abbreviated Switch	Purpose
--config-directory	-cd	Set the storage directory for the config files
--help	-h, -?	Show help information
--interactive	-i	Enable interactive mode
--key-directory	-kd	Set the storage directory for the config file encryption key
--verbose	-v	Output verbose errors

token

Retrieve an access token to use in your scripts.

```
tss token [options]
```

Switch	Abbreviated Switch	Purpose
--config-directory	-cd	Set the storage directory for the config files
--help	-h, -?	Show help information
--interactive	-i	Enable interactive mode
--key-directory	-kd	Set the storage directory for the config file encryption key
--verbose	-v	Output verbose errors

version

Display the version of the SDK client.

```
tss version [options]
```

Switch	Abbreviated Switch	Purpose
--config-directory	-cd	Set the storage directory for the config files
--full	-f	Display the full diagnostic version of the SDK client

Switch	Abbreviated Switch	Purpose
--help	-h, -?	Show help information
--interactive	-i	Enable interactive mode
--key-directory	-kd	Set the storage directory for the config file encryption key
--verbose	-v	Output verbose errors

Syncing with DevOps Secrets Vault

Overview

Secret Server can push its secrets to DevOps Secrets Vault by creating a secret based on the "DevOps Secret Vault Client Credentials" template, which holds the client credentials for a DevOps Secrets Vault tenant. Using the REST API, you can then register a DevOps Secrets Vault tenant in Secret Server. That tenant references that secret to push secrets to DevOps Secrets Vault at a set sync interval.

Behavior Test

You can manually push secrets to the DSV tenant, in addition to Secret Server checking for secrets to push to tenants on a timer. Secret Server will check for if a tenant needs updating every 30 minutes on the cloud or 10 minutes for an on-premises installation. Users are prevented from setting a tenant's sync interval to less than Secret Server's timed iteration because there would be no benefit to doing so. When Secret Server checks for secrets to be pushed to DSV, it only pushes secrets that have been changed since the last time they were updated in DSV. When a secret is pushed to DSV, its sync time is updated.

Fields

All secret fields are copied to DSV except for fields that are marked as "Hide On View." The notes field of a secret maps to the secret description in DSV. Files are Base64 Encoded, then sent to DSV. They are stored as encoded, and need decoding for use.

Setup in Secret Server

To configure pushing secrets to DSV:

1. Create a client in DSV. Save the client ID and secret that are generated when you created it. A DSV client is a container for a password.

> Note that before creating a client, you should create a DSV role - see DSV Role documentation. After that, you should use that role to create a DSV client - see DSV Client documentation. Then, create or edit the specific DSV policy that gives the role access to create DSV secrets on a specified path - see DSV Policies documentation.

2. Create a secret based on the DevOps Secrets Vault Client Credentials template to connect to DSV:
   - Type the name for the new secret in the **Secret Name** text box.
   - Type the DSV client ID in the **Client ID** text box.
   - Type the DSV password for authentication in the **Client Secret** text box. If you do not have one, you can create a new here by clicking the **Generate** button. Then, create or configure a client in DSV using the password.
   - Type the DSV tenant to connect to in the **Tenant** text box. A DSV tenant is your DSV cloud account and the rights to access it. Use the format: `https://<tenantname>.secretsvaultcloud.<region>` with the region being one of the following:
     - U.S. region: `com`
     - E.U. region: `eu`
     - APAC region: `au`
   - Click the **Site** dropdown list to select your Secret Server site.
   - Click the **Create Secret** button.
3. Go to **Admin > DevOps Vault**. The DevOps Secrets Vault Tenants page appears.
4. Click **Add New Tenant**. The Add New Tenant popup appears.
5. Type a descriptive name for the tenant in the **Tenant Name** text box. This can be anything you wish.
6. Click the **Client Secret** link to select the secret you created earlier in this instruction.
7. Click the **Sync Interval** list box to select how often you want Secret Server to push secrets to DSV for this tenant.
8. Click the **Save** button.

API Examples

Creating a DevOps Secrets Vault Tenant

Use a POST to `/api/v1/devops-secrets-vault/tenant` using the body below to create a tenant in Secret Server.

```
{
  "Data": {
    "secretId": { "value": 79, "dirty": true },
    "tenantName": { "value": "LJDevTenant", "dirty": true },
    "syncInterval": { "value": 60, "dirty": true },
    "active": { "value": true, "dirty": true }
  }
}
```

The secret ID is the client ID for the secret based on the DSV Client Credentials template. The Sync Interval is how often Secret Server checks if secrets needs to be pushed to DSV. Only secrets associated with active tenants are pushed to DSV. You are returned the tenant ID if the POST is successful.

Creating a Sync Map

Use a POST to `/api/v1/devops-secrets-vault/add-sync` using this body to map a secret to a DSV tenant:

```
{
"data": {
  "secretId": {
    "dirty": true,
    "value": 60
  },
  "dsvTenantId": {
    "dirty": true,
    "value": 1
  },
  "active": {
    "dirty": true,
    "value": true
  },
  "fieldNamesPath":{
    "dirty": true,
    "value": [
      "Demo","\$domain","qagreentest"
      ]
    }
  }
}
```

When the secret is mapped to a tenant, an initial sync immediately occurs. Following the initial sync, the secret is checked to determine if updates have been made when the sync Interval expires (making it "dirty") for the mapped tenant. If no changes have been made to the secret, then the secret is not pushed to DSV. You can reference fields from the secret to create the path in DSV. Secret Server will look for a `$`, then search for the following string as the "Field Slug Names" on page 1175 for the secret's template. The path in DSV follows this format: `/secrets/<DSV_secret_name>`.

Manually Syncing a Secret

Use a POST to `/api/v1/devops-secrets-vault/sync` to manually trigger a push to DSV for existing sync maps. The list of integers contains the SyncMapIds of the secret to tenant mapping, so you can control which secret is pushed to which tenant.

```
{
"data": [
  3, 4, 5
  ]
}
```

Listing DevOps Secrets Vault Tenants

List DSV tenants registered to Secret Server by running a GET to `/api/v1/devops-secrets-vault/tenant`. Query parameters accepted:

- `filter.nameSearch=`
- `filter.includeInactive=`

Getting a DevOps Secrets Vault Tenant's Details

View the details of a single tenant by specifying a tenant ID in a GET to `/api/v1/devops-secrets-vault/tenant/{tenantId}`.

Getting the Status of a Secret's Synchronization

View a secret's sync status by running a GET to `/api/v1/devops-secrets-vault/sync/status/{syncMapdId}`.

Getting a List of Secret Synchronization Statuses

View a list of secret sync statuses by running a GET to `/api/v1/devops-secrets-vault/sync/status`. Query parameters accepted:

- `filter.secretId=`
- `filter.includeInactive=`
- `filter.tenantId=`

Secret Server Release Notes

Secret Server release notes provide detailed information about the latest updates, enhancements, and bug fixes for Secret Server. These notes are essential for administrators and users to stay informed about new features, security improvements, and any changes that might affect their usage of the system. Each release note typically includes the release date, version number (for On-Premises versions), a summary of new features, detailed descriptions of enhancements, and a list of resolved issues.

Archived On-Premises versions are those that are no longer supported by Delinea. Archived Cloud versions are those over a year old.

For changes pushed to cloud between major releases, see the Secret Server Change Log.

Secret Server On-Premises Release Notes

These are the release notes for Secret Server On-Premises versions that are still supported by Delinea.

Secret Server 11.7.000061 Release Notes

Release Date: On-premises: February 10, 2025

Version Information

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.43.0

Protocol Handler: 6.0.3.33

> If your protocol handler version is 6.0.3.26 or lower, you must manually upgrade to a higher version. Automatic upgrades will not work for versions 6.0.3.26 or below. However, if your protocol handler version is 6.0.3.27 or higher, the automatic upgrade will function properly.

Improvements

548904	Improved: Performance updates for SSH Proxy when using an SFTP subsystem connection.
558857	Improved: New UI for selecting restricted lists on teams. Allows user to select and manage more than 60 lists.
560728	Improved: The activity items in the session recording list have been simplified and extended details now appear below the video when a user clicks.
585624	Improved: TOTP history icons on the secret settings tab are now always visible and do not require hovering to see.
595555	Improved: Downloading event subscriptions now includes the target column in the .csv file.
601841	Improved: The API call to create a group now has a flag to control behavior for pre-existing group names. The flag specifies whether the call should fail creating groups or if the API should generate a unique name by appending a suffix as it did before.
602146	Improved: UI of Platform Migration Center has been localized.
602588	Improved: Added additional logging to clarify the reasons why a secret was skipped for RPC.
603971	Improved: Report editor SQL runner now updates columns when the SQL columns are updated.
607689	Improved: Accounts with a heartbeat failure status on an associated secret, displaying an error chip in the network view, now display a "heartbeat failed" chip and message on the details tab.

609150	Improved: Added support for French (Canadian), Italian, Dutch, and Polish.
609271	Improved: Updated the API to handle the "Quote Tokens" setting for PowerShell script secret dependencies.
609274	Improved: Changed default for timeout to Platform from 100 seconds to 5 minutes. Made this into an advanced configuration option in case it needs to be adjusted in the future.
610216	Improved: Details pages for secret-associated discovery accounts now include localized detailed messages below the "heartbeat failed error" chip for all heartbeat failure statuses.
610768	Improved: Added sorting to the directory account grid under the discovery network view.
613946	Improved: Added a user preference to underline links.
614507	Improved: Added a text instruction for Step 2 of the Platform Integration Center.
615583	Improved: Favorite and recent added as login home options.
615589	Improved: Stub secret API endpoint will now include which fields map to the password changer.
615939	Improved: Secret Server now contains support for PostgreSQL account secrets. PostgreSQL account secrets can participate in heartbeat and RPC operations.
615740	Improved: Selecting a filter will now announce the filter description after reading the label (accessibility).
616742	Improved: Upon selecting a single select filter the selected option is now associated and announced (accessibility).
616744	Improved: Remove filter button now uniquely announces which filter will be removed instead of just "remove filter" (accessibility).

Fixed Issues

545095	Fixed: In a secret's card view, under the shared users section, the initials allowed into the small colored Icon for each user was reduced from three to two, as some combinations of three letters were long enough to overflow the icon.

548022	Fixed: Updating roles on a group will no longer impact secret template permissions. Prior to this change, the TemplateCreateSecret role would show on the roles tab and get removed upon update of roles.
548815	Fixed: REST API doc for getting token now indicates that the user name can include the domain.
548967	Fixed: Discovery take over created secrets with the same name even when "Allow duplicate secret names" is set to false.
584024	Fixed: Multi-line text will now formats correctly when exported to a .csv file.
592485	Fixed: A check was added to verify that at least one permission from Secret Server is available in Platform to ensure users are not locked out during migration.
594281	Fixed: There was an issue where the ITSM integration script failed to run, resulting in an 'Object reference not set to an instance of an object' error when the 'Ticket System Publicly Available' option was disabled and a specific site was selected. The script now executes properly regardless of this setting, ensuring ITSM integration works as expected.
595576	Fixed: Bulk reactivation of disabled secrets that have deactivated templates now works, consistent with activating them one by one with a disabled template.
603969	Fixed: Report editor layout clipping issues.
604097	Fixed: Increased the timeout period for discovery scans to two days, allowing longer-running scans to complete without being canceled.
605349	Fixed: Resolved error that prevented a restricted user from updating lists on a secret.
605427	Fixed: Addressed an issue where customers without outbound internet access might be unable to generate session recordings.
606973	Fixed: Updated SafeReportSqlChecker to prevent queries using "sysdatabases" and bypassing table exclusions by prepending "dbo."
607694	Fixed: Addressed an issue where some computer accounts showed a "computer scan error" chip in details but not in the network view grid.

607730	Fixed: Resolved a resilient secret issue where in-progress replication log summaries were truncated mid-sentence during large dataset replication causing the replication state to remain stuck as "partially successful" even when the process completed successfully. This fix ensures smoother replication workflows, even for environments with large datasets.
608089	Fixed: Corrected the column label in the Discovery Network View. Previously mislabeled as "Full Name," the column now correctly displays as "Account Total" for improved clarity and accuracy.
610542	Fixed: Addressed a timing issue that could cause localization keys to show instead of the actual text in Platform.
610595	Fixed: Addressed an edge case where computers from a disabled discovery source would still appear in the Platform inventory.
610621	Fixed: UserId update (instead of name) for internal vault accounts.
611124	Fixed: User's enabled status is no longer determined by licensing and permission during sync from Platform to Secret Server.
611863	Fixed: Azure Active Directory renamed to Microsoft Entra domain on the directory services grid following Microsoft's renaming guidelines.
612788	Fixed: Addressed issue with being able to correct expired Azure AD domain credentials for clients that are not yet using "sync secrets."
613129	Fixed: Corrected a typo on add scanners filter localization that showed a duplicate option instead of "Show all scanners."
613755	Fixed: Directory services log now indicates Microsoft Entra instead of Azure AD.
614044	Fixed: Force check-in now allows the user to get into a secret when the password change is failing, canceled, and retried.
614742	Fixed: Updated filter for secret favorites is now applied properly.
615582	Fixed: When using Platform with Secret Server, an Active Directory user using a connector that is removed from Platform could be re-enabled in Secret Server.
616739	Fixed: The "Add item" button on top of grids is now labeled "Add filter."

616745	Fixed: The multi-select filter is now properly defined as role=menuitemcheckbox and the search announces how many are selected (accessibility).
626850	Fixed: An issue where secret policies configured with the "Standard" approval setting (where editors and approvers do not need approval) were not applying correctly. Instead, the Require Approval Type field incorrectly displayed as No Approval Required, overriding the expected policy behavior. The policy now correctly applies to secrets, ensuring that approval requirements reflect the assigned policy settings.

Known Issues

None at this time.

Secret Server 11.7.000060 Release Notes

Release Date: On-premises: January 24, 2025

Version Information

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.43.0

Protocol Handler: 6.0.3.32

> If your protocol handler version is 6.0.3.26 or lower, you must manually upgrade to a higher version. Automatic upgrades will not work for versions 6.0.3.26 or below. However, if your protocol handler version is 6.0.3.27 or higher, the automatic upgrade will function properly.

Improvements

548904	Improved: Performance updates for SSH Proxy when using an SFTP subsystem connection.
558857	Improved: New UI for selecting restricted lists on teams. Allows user to select and manage more than 60 lists.
560728	Improved: The activity items in the session recording list have been simplified and extended details now appear below the video when a user clicks.
585624	Improved: TOTP history icons on the secret settings tab are now always visible and do not require hovering to see.

595555	Improved: Downloading event subscriptions now includes the target column in the .csv file.
601841	Improved: The API call to create a group now has a flag to control behavior for pre-existing group names. The flag specifies whether the call should fail creating groups or if the API should generate a unique name by appending a suffix as it did before.
602146	Improved: UI of Platform Migration Center has been localized.
602588	Improved: Added additional logging to clarify the reasons why a secret was skipped for RPC.
603971	Improved: Report editor SQL runner now updates columns when the SQL columns are updated.
607689	Improved: Accounts with a heartbeat failure status on an associated secret, displaying an error chip in the network view, now display a "heartbeat failed" chip and message on the details tab.
609150	Improved: Added support for French (Canadian), Italian, Dutch, and Polish.
609271	Improved: Updated the API to handle the "Quote Tokens" setting for PowerShell script secret dependencies.
609274	Improved: Changed default for timeout to Platform from 100 seconds to 5 minutes. Made this into an advanced configuration option in case it needs to be adjusted in the future.
610216	Improved: Details pages for secret-associated discovery accounts now include localized detailed messages below the "heartbeat failed error" chip for all heartbeat failure statuses.
610768	Improved: Added sorting to the directory account grid under the discovery network view.
613946	Improved: Added a user preference to underline links.
614507	Improved: Added a text instruction for Step 2 of the Platform Integration Center.
615583	Improved: Favorite and recent added as login home options.
615589	Improved: Stub secret API endpoint will now include which fields map to the password changer.

615939	Improved: Secret Server now contains support for PostgreSQL account secrets. PostgreSQL account secrets can participate in heartbeat and RPC operations.
615740	Improved: Selecting a filter will now announce the filter description after reading the label (accessibility).
616742	Improved: Upon selecting a single select filter the selected option is now associated and announced (accessibility).
616744	Improved: Remove filter button now uniquely announces which filter will be removed instead of just "remove filter" (accessibility).

Fixed Issues

545095	Fixed: In a secret's card view, under the shared users section, the initials allowed into the small colored Icon for each user was reduced from three to two, as some combinations of three letters were long enough to overflow the icon.
548022	Fixed: Updating roles on a group will no longer impact secret template permissions. Prior to this change, the TemplateCreateSecret role would show on the roles tab and get removed upon update of roles.
548815	Fixed: REST API doc for getting token now indicates that the user name can include the domain.
548967	Fixed: Discovery take over created secrets with the same name even when "Allow duplicate secret names" is set to false.
584024	Fixed: Multi-line text will now formats correctly when exported to a .csv file.
592485	Fixed: A check was added to verify that at least one permission from Secret Server is available in Platform to ensure users are not locked out during migration.
594281	Fixed: There was an issue where the ITSM integration script failed to run, resulting in an 'Object reference not set to an instance of an object' error when the 'Ticket System Publicly Available' option was disabled and a specific site was selected. The script now executes properly regardless of this setting, ensuring ITSM integration works as expected.
595576	Fixed: Bulk reactivation of disabled secrets that have deactivated templates now works, consistent with activating them one by one with a disabled template.
603969	Fixed: Report editor layout clipping issues.
604097	Fixed: Increased the timeout period for discovery scans to two days, allowing longer-running scans to complete without being canceled.

605349	Fixed: Resolved error that prevented a restricted user from updating lists on a secret.
605427	Fixed: Addressed an issue where customers without outbound internet access might be unable to generate session recordings.
606973	Fixed: Updated SafeReportSqlChecker to prevent queries using "sysdatabases" and bypassing table exclusions by prepending "dbo."
607694	Fixed: Addressed an issue where some computer accounts showed a "computer scan error" chip in details but not in the network view grid.
607730	Fixed: Resolved a resilient secret issue where in-progress replication log summaries were truncated mid-sentence during large dataset replication causing the replication state to remain stuck as "partially successful" even when the process completed successfully. This fix ensures smoother replication workflows, even for environments with large datasets.
608089	Fixed: Corrected the column label in the Discovery Network View. Previously mislabeled as "Full Name," the column now correctly displays as "Account Total" for improved clarity and accuracy.
610542	Fixed: Addressed a timing issue that could cause localization keys to show instead of the actual text in Platform.
610595	Fixed: Addressed an edge case where computers from a disabled discovery source would still appear in the Platform inventory.
610621	Fixed: UserId update (instead of name) for internal vault accounts.
611124	Fixed: User's enabled status is no longer determined by licensing and permission during sync from Platform to Secret Server.
611863	Fixed: Azure Active Directory renamed to Microsoft Entra domain on the directory services grid following Microsoft's renaming guidelines.
612788	Fixed: Addressed issue with being able to correct expired Azure AD domain credentials for clients that are not yet using "sync secrets."
613129	Fixed: Corrected a typo on add scanners filter localization that showed a duplicate option instead of "Show all scanners."
613755	Fixed: Directory services log now indicates Microsoft Entra instead of Azure AD.
614044	Fixed: Force check-in now allows the user to get into a secret when the password change is failing, canceled, and retried.
614742	Fixed: Updated filter for secret favorites is now applied properly.

615582	Fixed: When using Platform with Secret Server, an Active Directory user using a connector that is removed from Platform could be re-enabled in Secret Server.
616739	Fixed: The "Add item" button on top of grids is now labeled "Add filter."
616745	Fixed: The multi-select filter is now properly defined as role=menuitemcheckbox and the search announces how many are selected (accessibility).

Known Issues

None at this time.

Secret Server 11.7.000049 Release Notes

Release Date: On-premises: November 26, 2024

Version Information

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.39.0

Protocol Handler: 6.0.3.31

> If your protocol handler version is 6.0.3.26 or lower, you must manually upgrade to a higher version. Automatic upgrades will not work for versions 6.0.3.26 or below. However, if your protocol handler version is 6.0.3.27 or higher, the automatic upgrade will function properly.

Improvements

537407	Improved: Sorting by a secret field for which there are no secrets now indicates such instead of just displaying an unknown column.
541150	Improved: When the Radius two-factor authentication login page loads for Secret Server, focus is now set to the password box, so you do not have to click it before typing.
545215	Improved: Unix discovery sources are now enabled by default when created successfully.
551651	Improved: Secret template fields can now be passed-in as arguments to ticket system scripts.
554428	Improved: Notification messages now disappear from the unread screen once marked read.
554463	Improved: Audit logging for user and group changes.
569648	Improved: Updated secret export to make it more resilient for larger environments.

572584	Improved: Added security hardening report and warnings to discourage or prevent password type changers that rely on JScape libraries due to security vulnerability.
575586	Improved: Upgraded audit notes for secret RPC settings changes. Added RPC_SCHEDULE_UPDATED and RPC_AUTOCHANGE_UPDATED secret audit actions.
575589	Improved: Disaster Recovery feature now replicates Open LDAP domain sync settings.
578673	Improved: Localization updates on the scripting pages.
578682	Improved: Inbox templates, rules, and resource grids converted to latest component grid.
580088	Improved: Azure Active Directory domains now use sync secrets yet remain backwards compatible for existing Azure Active Directory domains that still directly store the domain's client id, client secret, and tenant ID.
580662	Improved: Expanding the folder row in a secret grid informs the user inline if they lack access to see the folder owners instead of redirecting to a full access denied error page.
582196	Improved: Added a setting to redirect all users to log in through Platform integration settings.
584486	Improved: Removed case sensitivity from scriptable discovery lookup calls to find OUs for a computer.
584579	Improved: Site sorting is now in alphabetical order in dropdowns, lists, and more.
584837	Improved: Secret audits now have the ability to filter by audit action type via a multiselect filter.
586768	Improved: SQL editor dialog is now the full viewport height and width.
586770	Improved: Discovery accounts bulk action checkboxes removed on some pages where they were not necessary
587667	Improved: Secret Server custom URL is now being sent to Platform as an accepted redirect URL.
588435	Improved: Converted grids on three Automatic Export pages (Audit, Log and Storage) to thy-data-view for consistent look, feel, and functionality, such as allowing Notes field to be fully viewed in right panel on row click rather than via a tooltip.
590053	Improved: We now only show synchronized directory service groups in the list if active.
590073	Improved: OU column is now available on Discovery Network view.
590111	Improved: Discovery log and computer scan logs filter updated to latest component for accessibility.
590933	Improved: Discovery now has full detail pages for Entra entities, and they also appear in the new larger panel.

591264	Improved: Discovery network view now opens a larger panel with more information including the last error message. The large panel supports wider screen sizes as well to give side-by-side viewing of the grid and panel.
591933	Improved: Search performance improved for typical users.
592094	Improved: Bulk sync now does one insert SQL for a list of users and not one per user.
592264	Improved: Added a precheck to ensure we don't try to migrate native/hybrid users that are missing their extended mapping, which likely would have resulted in strange duplication behavior
592499	Improved: Groups created in Platform now default to being set as "Migrated" for their migration state.
592713	Improved: Category list converted to latest grid component.
592715	Improved: Azure AD Domains now use sync secrets. As part of this, searching for a suitable secret template to serve as the sync secret must have Client Id, Client secret, and Tenant Id fields mapped for its secret type. This applies to the Azure App registration secret template out of the box.
593909	Improved: Attempting to log into an application account through the UI will no longer add a successful login audit event for that user. Successful and failed login attempts are still logged.
595158	Improved: Secret Server reports can now be saved as a shortcut on the Platform desktop.
595573	Improved: The opt-in step allows a user to add a new Platform tenant. It is the first step to integrate from Secret Server to Platform. This step is a part of the redesign of the Easy Move integration tool.
595730	Improved: The team-members list page has been updated for easier adding and removing of members. The entire domain inclusion option has been moved to the team general tab.
595732	Improved: Roles mapped to the "All Vault Users" group in Secret Server are now be mapped to the "Everybody" group in Platform after migration runs if the "All Vault Users" group is selected for that migration.
595945	Improved: Report list page heading is now semantically correct.
595948	Improved: Secret quick-access page heading is now semantically correct.
596411	Improved: Updated Platform migration prechecks by adding new information precheck type.
596413	Improved: QuantumLock audit grid updated to latest component.
596695	Improved: Updated BouncyCastle.Cryptography package to a non-vulnerable version.

596927	Improvement. Added the ability to prompt for site selection when launching an SSH session through a DE proxy.
597484	Improved: You can now pick values less than four hours in the minimum Platform sync configuration.
597621	Improved: When on prem has a cluster issue, the license pages are now accessible from any web node.
599089	Improved: A new user experience setting has been added called, "Separate Secret Audit for Comment." When this is true, a secret that requires comment will have an additional audit entry with an action of "Comment." This allows a secret to be commented on and checked out but not viewed. Without this setting the secret audit will show a "View" action with the comment text and then the checkout. Now you will see "Comment," "Checkout," and then "View" only once they have actually viewed the secret.
599421	Improved: Creating a new Active Directory discovery source is now full page instead of in a modal.
599481	Improved: ITDR Service Accounts are now handled via messaging in Secret Server.
599717	Improved: Session monitoring routes were updated to no longer include /admin.
599937	Improved: Converted the disaster recovery audit to the new grid component.
600029	Improved: Converted the disaster recovery Log to the new grid component.
600137	Improved: Platform users metadata replicates fully to a disaster recovery replica instance.
600425	Improved: Converted directory services domain audit to the new grid component.
600639	Improved: Converted the export-import settings audit page to thy-data-view.
600664	Improved: For Secret Server Cloud customers integrated with Platform, disabled the underlying "Create Groups During Synchronization" setting, which was previously able to be enabled, and already disabled for the vast majority of customers. This setting would automatically create domain groups during synchronization, which we now require to be specifically created. Platform Cloud groups are already automatically created and this behavior is unaffected.
600797	Improved: Converted the secret-erase list grids to the new grid component.
601395	Improved: Discovery pages are now accessible at /discovery instead of /admin/discovery. This change also decreases the pack size of the admin and discovery modules.
601455	Improved: The toggle expand and favorite star in the global right panel widget now properly have aria-labels, aria-controls, and aria-expanded tags.

601456	Improved: Service users created in Platform are now mapped to application accounts in Secret Server.
601459	Improved: ITDR account created via OAuth token or messaging from identity.
602245	Improved: Updates to the opt-in flow: Platform region field is now read only by default, and regionEditable=true|false query parameter is available to override the default.
602371	Improved: Added security hardening report and warnings to discourage or prevent password type changers that rely on JScape libraries due to security vulnerability.
602443	Improved: Users can now navigate to and from licenses page to clustering pages.
602663	Improved: Resilient secrets now handle additional conflict scenarios between the source and replica.
602778	Improved: Updated IBM dark mode left navigation for better accessibility experience.
603115	Improved: Reduced Redis calls for many operations for performance purposes.
603328	Improved: Added logging to capture specific failure codes from Entra ID when performing heartbeat.
603363	Improved: Refresh button was added to the migration audit tab of Platform integration center.Improved: Filters implemented on migration audit tab of Platform integration center.
603364	Improved: A filter of object type was added to the log tab of the Platform integration center.
603978	Improved: Password changers grid now includes columns for can edit and Secret usage count. Some additional filters were also added.
603980	Improved: VaultBroker updating a URL requires a valid connection with the Secret Server instance before allowing updates.
604471	Improved: Added security hardening report and warnings to discourage/prevent password type changers that rely on JScape libraries due to security vulnerability.
604568	Improved: Updated UI to only make one call to the "enable unified mode" endpoint.
609745	Improved: Performance of launcher session cleanup
609923	Improved: Added a check to Platform configuration in Secret Server that prevents synchronization of Platform data (users, groups, roles, etc.) to Secret Server while migration from Secret Server to Platform is running.

Fixed Issues

451250	Fixed: Addressed an issue where, when editing a user, the Duo multifactor authentication option would be missing when Radius and Duo had already been configured and the user had the Administer Users permission. Fixed: Addressed an issue where saving a User with the Duo Multifactor authentication option would throw an error when Radius and Duo had already been configured and the user had the Administer Users permission.
477012	Fixed: "What Secrets have failed heartbeat?" no longer shows secret with a failed heartbeat when their template has heartbeats turned off.
513832	Fixed: Sorting issue on the Groups tab in user management
514188	Fixed: Issue with MobaXterm launches and multiple credential-save prompts.
517513	Fixed: Not being able to add XML files where the key file folder is located because DPAPI read all XML files in that folder and did not find the correct and expected format, which caused an error and disconnected from the Secret Server instance.
541011	Fixed: Guide dialogs now properly focus the guide and trap tab.
543702	Fixed: Session recording processing no longer causes a false session recording view event
544916	Fixed: Corrected Cipher Suite connection issue to AWS EC2 instances with public keys only.
544998	Fixed: SSH key expiration in label description is displayed correctly now.
547577	Fixed: Event subscription language resource corrected for "Engine" and "Export Secrets" events.
561895	Fixed: Workflow Approval email "View this item" contained incorrect link, not directing users to a page where they could approve the request
563019	Fixed: "Minimum Required Character Count Rules" on password requirements reverts when updating other things on password requirements.
563367	Fixed: Addressed an issue where the video recording tab would display for session recordings that were keystroke only.
563529	Fixed: UI issue on report schedule page where unchecking "send email" for report distribution blocked saving.
567824	Fixed: SAML Log no longer opens a dialog and a preview panel.
569536	Fixed: Issue were using Secret Server SDK 1.5.7 or earlier after upgrading to 11.7 gave an "Object reference not set to an instance of an object." error when trying to retrieve a secret. The fix appears in version 1.5.9.

570798	Fixed: In the SDK Client Management > Client Onboarding Page, when viewing a user with an onboarding key required, if the key is visible in the side panel and you select another user with a key required, the key that is shown now updates to be the key associated with the most recently selected user.
571356	Fixed: Event queue not clearing. Adjusted the location of and query of the EventQueue cleanup process.
572635	Fixed: RPC errors for SAP template secrets, which were occurring with SAP "systemuser" user types, even when using a privileged account.
575347	Fixed: IBM code editor resources properly shared to allow editor to load.
575767	Fixed: Removed Thycotic.Ihawu.UnitTests.Web and Thycotic.Ihawu.UnitTests.Web.Rest from repository, which have not been active since 2018.
575896	Fixed: Enabled Entra ID Password Changer to appropriately handle heartbeat on accounts where MFA is applied through a Conditional Access Policy.
576164	Fixed: Added DelayBackgroundStartupMilliseconds to fix a race condition during the PKCS #11 login after integrating with the Entrust HSM. This will delay the background workers so the web node can log in to the PKCS #11 library first.
580552	Fixed: Group sharing with Secret Server secrets for groups would not trigger a sync for all existing users in Secret Server.
581752	Fixed: Custom report names with double quotations in them no longer throw an error when downloading a report to a .csv and will download successfully. The .csv will not contain the quotations, but they will remain in the name on Secret Server.
581802	Fixed: Some edge cases related to Platform Federation could result in a group losing its members in Secret Server.
585612	Fixed: A duplicate role assignment during migration was fixed so that the migration matches what was in Secret Server .
586300	Fixed: When a date is downloaded from a grid, it now properly formats according to the selected download date format.
586526	Fixed: Card and grid mode "last connected" field now shows the correct time and match.
586528	Fixed: Resolved an issue where attempting to use a Session Connector launcher with "Open with Remote Access" would throw an error when attempting to launch
587596	Fixed: NVDA now displays the correct labels for the tree component.

587768	Fixed: A bug where the password changing field of discovery rules would not update. The password changing settings now persists for all discovery rules.
588849	Fixed: Addressed Null Ref errors in SyncSessionToPlatformMessage.
588873	Fixed: Corrected impacts on user access by enabling or disabling DTC.
589002	Fixed: Token name-wrapping issue fixed on secret dependency dialog.
589194	Fixed: A bug where the source field of discovery rules would not update. Source settings now persist on all discovery rules.
589245	Fixed: When Secret Server was integrated with Platform and a federated user logs and connects to an AD user in Secret Server, the Platform sync could remove AD groups (from Secret Server AD Synchronization) because that was not a supported configuration. Now, we prevent the removal of AD groups from a Secret Server user during Platform synchronization if the connected Platform user source is Federation. Platform synchronization of groups when a Platform user source is Active Directory/connector and the Secret Server user is an Active Directory user will work as before. We are doing this as harm reduction until the configuration in Platform is set up to be compatible with supported scenarios.
589332	Fixed: Creating an access request with custom dates or times displayed an incorrect warning or had incorrect dates or times when approved.
589821	Fixed: Addressed a very rare edge case where a synchronized AD or Azure AD group flagged as "SynchronizeNow" that is also inactive could block synchronization from running indefinitely.
589824	Fixed: Issue where Entra ID accounts could be mistakenly identified as directory accounts in the discovery network view.
589974	Fixed: Issue that prevented creating empty discovery sources.
589977	Fixed: Addressed red banner issue on session monitoring page when using v2 grid filters.
590058	Fixed: Azure AD synchronization did not handle groups with null ADGuid fields. Added filter criteria in Thycotic.ActiveDirectory to filter out cases causing errors.
590449	Fixed: An issue where active users in inactive domains (an exceedingly rare edge case that we do not natively support) caused groups to fail importing valid users.
591870	Fixed: Migration work now considers users that have come from Platform (either Platform Native or Hybrid users) when migrating. It was indirectly ignoring them before under certain circumstances
591954	Fixed: Issue with connections remaining open with Windows local account RPC.

592169	Fixed: A duplicate group was created when Platform syncs back to Secret Server after a group is migrated.
592877	Fixed: The product link for installing browser extensions led to the documentation home rather than the proper page. Clicking OK now takes you to the correct page, "Installing Browser Extensions" on page 709.
592981	Fixed: Password dictionary uploads no longer fail due to Unix line endings.
593023	Fixed: Saving ticket system as publicly available now saves properly.
593302	Fixed: Changed Entra ID discovery scanner so it returns UPN for account name instead of display name.
593348	Fixed: Entra ID discovery can now identify members of a role who are assigned to that role through a group.
593359	Fixed: Issue that prevented Entra ID Roles from being automatically scanned by discovery.
593531	Fixed: In Platform, the browse all link in the folder tree that appeared after 1000 folders were shown was missing /vault in the URL.
593702	Fixed: An issue where the Entra ID discovery scanner flow could not be applied from the dialog that appears when creating an Entra ID discovery source.
593947	Fixed: When you launch a secret that requires checkout, you are no longer redirected to the secret detail page.
594073	Fixed: Discovery scanner text had a typo in the scanner CID notation example text.
594094	Fixed: An issue that prevented a resilient secrets (DR) replica from updating Secret Server after the source has been updated with 11.7.31.
594680	Fixed: Typo in Secret Erase Request.
594732	Fixed: DE Vulnerabilities cleaned up and verified removed.
595541	Fixed: Typo in bulk record selection dialog.
595553	Fixed: Some single-edit dialog fields were not properly linked to their label.
595556	Fixed: Inbox breadcrumb was going to a 404 error and has been removed.
595758	Fixed: Secrets with checkout and require comment will now have a combined option in the secret grid options menu for a secret. You can configure these to be separate as prior to this release in admin/user experience.

596124	Fixed: In the SDK Client Management > Client Onboarding Page, when viewing a user with an onboarding key required, if the key is visible in the side panel and you select another user with a key required, the key that is shown now updates to be the key associated with the most recently selected user.
596363	Fixed: Directory services will no longer appear in search when running in Platform.
596370	Fixed: Discovery computer scan results now properly defaults to showing the last hour.
596663	Fixed: An issue where certain usernames were unable to automatically sudo with SSH Proxy.
596721	Fixed: Resolved a case where event pipeline activity records were not being cleaned up according to the retention settings.
596851	Fixed: Secret template fields grid in reorder mode had an empty column heading.
597021	Fixed: Resolved an edge case of unmigrated users from Secret Server to Platform and ability to change usernames.
597482	Fixed: Addressed issue where a secret policy with an inactive user in an approver group could cause downstream issues when modifying related secrets.
598013	Fixed: Addressed a case where URI's were compared before normalization and canonicalization potentially leading to over matching against the approved list. Added additional validation that the downloaded installer's batch file was in the expected format.
598083	Fixed: An issue where failed video conversions would not clean up temp files.
599034	Fixed: Resolved accessibility issues on workflow and custom SSH cipher suites.
599078	Fixed: Corrected validation on tenant customization page.
599386	Fixed: Issue with updating workflow step names.
599798	Fixed: Folder searches now work when search text is in uppercase.
599966	Fixed: Platform/SSC to Onprem SS DR was unable to login as a Platform user. Added code to copy the custom URL from Secret Server every five minutes and send it over to platform as a valid redirect URL
600132	Fixed: An issue where users were unable to create secret policy via API with jumpbox site ID not set.
600415	Fixed: Entra ID heartbeat can now handle accounts that are pending MFA Enrollment. Added enhanced error handling to Entra ID account heartbeat.
602238	Fixed: IBM code editor resources properly shared to allow editor to load.

602673	Fixed: Two logic errors (one in the Easy Move path and one in the External User Mapping path) that were causing a null ref when trying to create a new user incorrectly. Domain users that are disabled by AutomaticUserManagement will no longer incorrectly cause a duplicate local user to be generated during migration and no null ref audit will be generated.
603148	Fixed: Addressed an issue where an exceptionally large foreign key in tbStatusMessage could cause errors when inserting records.
605472	Fixed: Addressed an issue that could prevent user creation and mapping of users from Platform when pre-existing users had been disabled by automatic user disabling.
610193	Fixed: Issue for creating vendor users when instance is already at maximum user licensed count.

Known Issues

None at this time.

Secret Server 11.7.000031 Release Notes

On-premises: August 20, 2024

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.33.0

Protocol Handler: 6.0.3.29

> If your protocol handler version is 6.0.3.26 or lower, you must manually upgrade to a higher version. Automatic upgrades will not work for versions 6.0.3.26 or below. However, if your protocol handler version is 6.0.3.27 or higher, the automatic upgrade will function properly.

New Features

Entra ID Discovery

We are excited to introduce Entra ID discovery in Delinea's Secret Server! This enhancement expands our current discovery feature by adding support for Microsoft's Entra ID, alongside our existing AWS and GCP discovery types.

With Entra ID discovery, Secret Server can now scan Microsoft Entra ID for roles and users, importing users as secrets based on the Entra ID User Account template. This completes the suite of features necessary for Secret Server to discover and manage accounts from Microsoft's Entra ID.

Bug Fixes, Changes, and Enhancements

Bug Fixes

Fixed “Secret Erase” translation in some non-English languages.

Fixed a bug where distributed engines ignored WinRM quota limits.

Fixed a bug with running disaster recovery data replication from an older source to a newer replica.

Fixed a UI issue with discovery Import where the button would not respond due to validation against hidden fields.

Fixed an issue during Platform group synchronization where groups with long names would cause an error.

Fixed an issue so users and roles now always show in SSC, even if in unified mode, but still are hidden when using Platform.

Fixed an issue that prevented resilient secrets (DR) replica from updating Secret Server after the source has been updated with 11.7.31.

Fixed an issue to restore configurability of secrets associated with custom launchers.

Fixed an issue where enabling QuantumLock on a secret threw the error “The partner transaction manager has disabled its support for remote/network transactions.”

Fixed an issue where stored data growth impacted proxy sessions. The secret session table is now managed and part of the supported tables of the data retention feature. Secret session records are now truncated in accordance with the existing data retention configuration. Please make sure to review your organization's Data Retention “Max Record Age” settings.

Fixed an issue where the “All time” filter on the inbox might not show all results.

Fixed an issue where users with MFA enabled would be incorrectly sent to the home page on login, instead of the page they were attempting to access.

Fixed an issue where video conversion failed due to SQL deadlock

Fixed content security policy fields for frame-ancestors.

Fixed incorrect access checks concerning reports.

Fixed incorrect Secret search totals when filtering by multiple templates.

Fixed issue where the “su -id” command was failing when the user did not have access to view the password for the secret they were elevating to.

Fixed issue where the maximum log Length was not used to truncate the tbSystemLog.

Fixed issue with “What folder permissions exist” report. Groups with no active users now properly included on the report

Fixed main navigation alignment issues.

Fixed ServiceNow allowed status validation over distributed engine.

Fixed the “view detail” link on the user detail panel.

Fixed The folder tree is now updated when unlimited admin mode is toggled.

Fixed timeouts for large amounts of data—paging for user audits is now done in the database.

Fixed: “Minimum Required Character Count Rules” on password requirements reverts when updating other things on password requirements.

Fixed: A user that did not have the “view launcher password” role permission was unable to create a secret that had a required password because the password field was hidden.

Fixed: About page links not working.

Fixed: Added null checks for username.

Fixed: Added support for Cisco devices when using a question mark after the command or partial command. This allows Cisco to work as normal, while not allowing the blocked commands.

Fixed: Addressed an issue where a launcher type field that was replicated via resilient secrets would not function with all prompt-able field names.

Fixed: Addressed one scenario where a backend process that publishes session information would error.

Fixed: Adjusted secret overview tab to not use a banner for heartbeat failed.

Fixed: Adjusted Secret tab pending password change status to be a chip instead of a banner.

Fixed: Audit handler was missing the “View Configuration Unlimited Admin” permission as an option.

Fixed: Authentication errors are now 401s for API requests and in Platform.

Fixed: Broken “view detail” link on the user detail panel.

Fixed: Creating an access request with custom dates/times displayed an incorrect warning or had incorrect dates/times when approved.

Fixed: Customers who had Easy Move to Platform had duplicate groups created in Secret Server and the existing permissions from the original Secret Server group were not honored. It now disables this new duplicate group and connects the original group to the Platform group as originally expected.

Fixed: Discovery runtime summary information is now correctly accessible for screen readers.

Fixed: Distributed engine now respects the MaxShellsPerUser setting for PowerShell tasks. If the setting is set, engine will throttle tasks that leverage PowerShell and requeue messages that are over quota.

Fixed: During forwarding of inventory data from discovery in Secret Server to Platform inventory, with large amounts of computers, the processes could time out. Made the database calls more efficient and the process no longer times out.

Fixed: Extended the Migration Center to migrate all active roles.

Fixed: Folder path now shows when specified in secret import preview.

Fixed: Heartbeat listed as “pending” when the heartbeat is actually disabled. This occurred when the pending status did not resolve before the secret was disabled.

Fixed: Improved “Regenerate Platform Credentials” to attempt to forward credentials to connected Secret Server Cloud automatically (behind feature flag).

Fixed: Improved compatibility with Windows high contrast mode.

Fixed: In some scenarios only the first 30 subfolders were loaded on initial load for a single folder.

Fixed: In some scenarios the folder tree would not auto-expand when linking directly to a folder.

Fixed: Left navigation expand/collapse toggle incorrectly labeled for screen readers.

Fixed: Login SSH key menu showing properly in cloud when configured.

Fixed: Mobile logo now displaying properly.

Fixed: Most KB links now point to docs.delinea.com instead of delinea.center for redirects to the documentation article.

Fixed: Newer versions of Safari can now play session recordings in Platform.

Fixed: Pinned folders now re-root the tree to the selected pinned folder.

Fixed: Reduced situations where a check-in error could occur when already checked-in.

Fixed: Removed links to legacy create discovery wizard pages.

Fixed: Resolved an issue that caused SAML logins to fail, resulting in a rollback of the previous update.

Fixed: Resolved an issue where approvals that cross a day threshold from UTC could not be requested.

Fixed: Resolved secret permission issue when many user and groups had been selected and only the 60 were saved when edited again. Resolved for teams selection as well.

Fixed: RPC errors for SAP template secrets, which were occurring with SAP “system user” user types, even when using a privileged account during the RPC.

Fixed: Searching in all secrets now shows the full folder path for folder search results.

Fixed: Secret Key rotation failed with the error “Thycotic.AppCore.Cryptography.MacMismatchException: Exception of type 'Thycotic.AppCore.Cryptography.MacMismatchException' was thrown.”

Fixed: Secret password compliance is now calculated when a password is updated to empty and the password is not required. Prior to this, the secret would maintain the compliance flag that was calculated when the password had a value. A password with some characters might fail compliance, but if there is no password and it is not required, then it is compliant.

Fixed: ServiceNow integration could fail with a misleading error due to a space in the domain name.

Fixed: Site name now wraps instead of truncating on the “sites and engines” page so you can read the whole site name.

Fixed: SQL report editor is now properly announced for accessibility.

Fixed: SSH keep-alives sent to the proxy are now relayed to the endpoint server.

Fixed: Suggested secret template toggle, when creating an inline secret from new discovery source, is now more closely positioned to the template list to be more clear.

Fixed: Teams group membership removed when more than 60 items in Team.

Fixed: The SSH key-expiration configuration value now displays correctly.

Fixed: Thycotic One Login Link.

Fixed: Unlimited admin mode audit dialog box is now correctly aligned.

Fixed: Updated all the logs to be warnings and information and to state whether they retried or not.

Fixed: Updated Discovery Network view to better handle extremely large record numbers.

Fixed: Updated the distributed engine service to persist the current the web-proxy.config file upon update. When upgrading to version from version 8.4.29.0 or lower, the web-proxy.config will be overwritten, but any upgrades afterwards will preserve it.

Fixed: User username link was sometimes unusable. It is is no longer a link. View details link is in menu and preview panel.

Fixed: When viewing folder targets for event pipeline policies the full path is now shown.

Fixed: when viewing the access-request inbox, the request start date and requested date were transposed.

Changes

Change: Admin breadcrumb renamed to Settings.

Change: Corrected license expiration banner link.

Change: Platform now specifies Secret Server configuration.

Change: Removed the color mode toggle from the top navigation as it is available under user preferences.

Change: RequirePlatformMfa field is now deprecated.

Change: The delinea.vault/secretserver/access permission has been removed. This no longer controls Secret Server access for Platform users.

Change: The SSL menu item is removed as it is not an option that can be modified in cloud.

Change: User list detail link added back based on user feedback.

Enhancements

Enhancement: Added “RPC PRIVILEGED SECRET UPDATED” and “RPC PRIVILEGED SECRET REMOVED” events to audits.

Enhancement: Added a “Clear cached AD credentials” button in cloud.

Enhancement: Added a “test syslog” button to syslog pages in configuration.

Enhancement: Added a direct link for launching connection manager.

Enhancement: Added a setting to redirect all users to log in through Platform integration settings.

Enhancement: Added AIX support for SSH Proxy su automatic password entry.

Enhancement: Added an OOB RPC template for Okta. Okta requires an “Generic API” secret as the RPC privileged account.

Enhancement: Added an OOB RPC template for ServiceNow. ServiceNow requires an account to have Admin or write permissions to the password field, or an account with those permissions as its RPC privileged account to change the password.

Enhancement: Added DSV links to the Platform settings page.

Enhancement: Added landing page for when the user is unable to access Secret Server instead of showing banners.

Enhancement: Aria label added to inline secret-preview copy buttons. Main search category toggles now keyboard accessible.

Enhancement: Associated secrets will now show “No Access” in the secret name if you do not have access to it.

Enhancement: Converted grids on 3 Automatic Export pages (Audit, Log and Storage) to thy-data-view for consistent look/feel and functionality, such as allowing Notes field to be fully viewed in right panel on row click rather than via tooltip.

Enhancement: Converted key management to the latest design and added a verification checkbox confirmation step.

Enhancement: Creating secrets in REST API now accepts optional parameters for privileged secret ID and associated secrets.

Enhancement: Heartbeat and password-compliance notices now use chips instead of banners.

Enhancement: Improved startup logging for distributed engines.

Enhancement: New import secret page allows you to import when global setting requires that secrets are in folders.

Enhancement: On premises now shows a diagnostics section under settings in the left navigation panel.

Enhancement: The left navigation folder tree now expands on focus to show longer folder names.

Enhancement: Updated password compliance label to a chip.

Enhancement: Updated Putty to version 0.81. Updated version addresses several Putty vulnerabilities, including the Terrapin vulnerability.

Enhancement: Updated Redis library for improved Redis operations.

Enhancement: Updated the server nodes page.

Enhancement: Updated the user profile menu to have more consistent styling and include links to the account details page.

Enhancement: Updated user experience for adding custom logos to Platform instances.

Enhancement: Updated user sorting to cover 2FA.

Enhancement: When a Secret Server is integrated with a Platform tenant, any Platform cloud groups are now automatically and quickly be created in Secret Server to be available for permission delegation.

Secret Server 11.7.000016 Release Notes

On-premises: June 12, 2024

11.7.000016 resolves a problem with older versions of Rabbit MQ (before 3.10) impacting DE version 8.4.31 that shipped with 11.7.000015. We pulled 11.7.000015 to resolve this issue. All other features, enhancements and bug fixes from 11.7.000015 are now in 11.7.000016

For convenience, we repeated the Features, Enhancements, and Bugs sections from the 11.7.000015 release notes here.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.32.0

Protocol Handler: 6.0.3.28

If your protocol handler version is 6.0.3.26 or lower, you must manually upgrade to a higher version. Automatic upgrades will not work for versions 6.0.3.26 or below. However, if your protocol handler version is 6.0.3.27 or higher, the automatic upgrade will function properly.

Features

Entra ID Secret Template for RPC

Secret Server has supported Azure AD remote password changing for several years, this overhaul creates a new password changer and template, Entra ID, that uses Oauth application credentials as a privileged account to change a user password. Entra ID is Microsoft's comprehensive cloud-based identity and access management solution that helps organizations securely manage identities and access across their Microsoft services and applications. Our password changer and template support MFA and conditional-access policies and does not require PowerShell.

Enhancements

- Enhancement: Updated PuTTY to version 0.81. The new version addresses several PuTTY vulnerabilities, including the Terrapin vulnerability.
- Enhancement: Added AIX support for SSH Proxy su automatic password entry.
- Enhancement: Added the same-site attribute to browser cookies, which is a security feature that helps prevent cross-site request forgery (CSRF) attacks. Same Site attribute value was set to lax to create a balance between security and usability.
- Enhancement: Increased back-end performance of event queue processing when there are a lot of inbox rules.
- Enhancement: Security issue contact instructions are now available at ./well-known/Security as specified in RFC9116.
- Enhancement: Significantly improved the performance of secret searches when using displayed secret fields.
- Enhancement: Updated Secure Blackbox to latest version. Secure Blackbox FIPS support was updated in documentation.
- Enhancement: Updated SSH functionality through Secure Blackbox to address Terrapin.

Bug Fixes

- Fixed "Secret Erase" translation in some non-English languages.
- Fixed a critical security vulnerability in the SOAP webservice.
- Fixed a policy validation issue that occurred when using a $itemvariable.variablename in schedule pipeline minutes.
- Fixed a UI issue where some site connectors were incorrectly showing as disabled.
- Fixed a visual bug when checking out a secret.
- Fixed an issue were a command would fail to enter vi or vim mode and would allow blocked commands. Also fixed an issue where using su before vi or vim would fail and would allow blocked commands.
- Fixed an issue where a ticket number was not present in SIEM logging.
- Fixed an issue where an error dialog appeared when adding a dependency with associated secrets.
- Fixed an issue where deleting computers from the discovery network view failed to show a confirmation dialog box before continuing.

- Fixed an issue where Handling secrets that fail heartbeat/password changes when using a PowerShell script threw a MaxShellsPerUser exception. For heartbeat: Added a new heartbeat status called "NeedsImmediateRetry" to bypass the secret-template retry interval. For Password Change: Ensured the retry attempts are not increased after failure.
- Fixed an issue where launching a secret from the new search would launch the first secret from the results returned, not the selected secret.
- Fixed an issue where OAuth parameters were not validated. The OpenIdConnect flow has been adjusted to validate the redirection URI.
- Fixed an issue where removing fields from discovery scan templates threw a disableField error.
- Fixed an issue where searching for a quotation mark could cause an error.
- Fixed an issue where secret export/import links in the All Settings Category view were missing.
- Fixed an issue where users other than owners could view TOTP backup codes.
- Fixed an issue where users with MFA enabled would be incorrectly sent to the home page on login, instead of the page they were attempting to access.
- Fixed an issue where IWA prevented DR sync calls from being processed correctly.
- Fixed an issue with adding discovery sources that match the domain of a current secret.
- Fixed an issue with key utilization within SOAP and REST API token generation.
- Fixed and issue where toggling a favorite secret triggered a grid refresh.
- Fixed issue where the "su -id" command was failing when the user did not have access to view the password for the secret they were elevating to.
- Fixed issues that could cause incorrect group or user interactions between Secret Server and Platform. We corrected an issue with Platform group synchronization that would not correctly add all group memberships when synching over 1000 groups.
- Fixed some issues with easy-move edge cases and system display.
- Fixed timeouts for large amounts of data—paging for user audits is now done in the database.
- Fixed unclear RPC logging. Updated the log message to clearly indicate when a password sets the next run time and is not doing a change attempt.
- Improved the placement of Secret Server user admin and role links. They are now on the top level in All Settings under the category header.
- Improved the UI for SSH cipher pages.

Secret Server 11.7.000015 Release Notes

On-premises: May 22, 2024

> Out of an abundance of caution, we are temporarily pulling down SS version 11.7.15 (on-premises) to resolve a problem with older versions of RabbitMQ (before 3.10) impacting DE version 8.4.31, which is shipped with SS 11.7.15. If you still require version 11.7.15, please contact Delinea Support to assist you further. You may install it as long as you do not upgrade the DE to version 8.4.31. We are working on resolving this issue and releasing an update within the next two weeks. This issue does not impact Secret Server Cloud.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.31.0

Protocol Handler: 6.0.3.28

> If your protocol handler version is 6.0.3.26 or lower, you must manually upgrade to a higher version. Automatic upgrades will not work for versions 6.0.3.26 or below. However, if your protocol handler version is 6.0.3.27 or higher, the automatic upgrade will function properly.

Features

Entra ID Secret Template for RPC

Secret Server has supported Azure AD remote password changing for several years, this overhaul creates a new password changer and template, Entra ID, that uses Oauth application credentials as a privileged account to change a user password. Entra ID is Microsoft's comprehensive cloud-based identity and access management solution that helps organizations securely manage identities and access across their Microsoft services and applications. Our password changer and template support MFA and conditional-access policies and does not require PowerShell.

Enhancements

- Enhancement: Updated PuTTY to version 0.81. The new version addresses several PuTTY vulnerabilities, including the Terrapin vulnerability.
- Enhancement: Added AIX support for SSH Proxy su automatic password entry.
- Enhancement: Added the same-site attribute to browser cookies, which is a security feature that helps prevent cross-site request forgery (CSRF) attacks. Same Site attribute value was set to lax to create a balance between security and usability.
- Enhancement: Increased back-end performance of event queue processing when there are a lot of inbox rules.
- Enhancement: Security issue contact instructions are now available at ./well-known/Security as specified in RFC9116.
- Enhancement: Significantly improved the performance of secret searches when using displayed secret fields.
- Enhancement: Updated Secure Blackbox to latest version. Secure Blackbox FIPS support was updated in documentation.
- Enhancement: Updated SSH functionality through Secure Blackbox to address Terrapin.

- Enhancement: Added new permission Migrate Data to Platform. This permission will be automatically applied to roles that contain both the Administer Users and Administer Platform Integration role permissions.

Bug Fixes

- Fixed "Secret Erase" translation in some non-English languages.
- Fixed a critical security vulnerability in the SOAP webservice.
- Fixed a policy validation issue that occurred when using a $itemvariable.variablename in schedule pipeline minutes.
- Fixed a UI issue where some site connectors were incorrectly showing as disabled.
- Fixed a visual bug when checking out a secret.
- Fixed an issue were a command would fail to enter vi or vim mode and would allow blocked commands. Also fixed an issue where using su before vi or vim would fail and would allow blocked commands.
- Fixed an issue where a ticket number was not present in SIEM logging.
- Fixed an issue where an error dialog appeared when adding a dependency with associated secrets.
- Fixed an issue where deleting computers from the discovery network view failed to show a confirmation dialog box before continuing.
- Fixed an issue where Handling secrets that fail heartbeat/password changes when using a PowerShell script threw a MaxShellsPerUser exception. For heartbeat: Added a new heartbeat status called "NeedsImmediateRetry" to bypass the secret-template retry interval. For Password Change: Ensured the retry attempts are not increased after failure.
- Fixed an issue where launching a secret from the new search would launch the first secret from the results returned, not the selected secret.
- Fixed an issue where OAuth parameters were not validated. The OpenIdConnect flow has been adjusted to validate the redirection URI.
- Fixed an issue where removing fields from discovery scan templates threw a disableField error.
- Fixed an issue where searching for a quotation mark could cause an error.
- Fixed an issue where secret export/import links in the All Settings Category view were missing.
- Fixed an issue where users other than owners could view TOTP backup codes.
- Fixed an issue where users with MFA enabled would be incorrectly sent to the home page on login, instead of the page they were attempting to access.
- Fixed an issue where IWA prevented DR sync calls from being processed correctly.
- Fixed an issue with adding discovery sources that match the domain of a current secret.
- Fixed an issue with key utilization within SOAP and REST API token generation.
- Fixed and issue where toggling a favorite secret triggered a grid refresh.
- Fixed issue where the "su -id" command was failing when the user did not have access to view the password for the secret they were elevating to.

- Fixed issues that could cause incorrect group or user interactions between Secret Server and Platform. We corrected an issue with Platform group synchronization that would not correctly add all group memberships when synching over 1000 groups.
- Fixed some issues with easy-move edge cases and system display.
- Fixed timeouts for large amounts of data—paging for user audits is now done in the database.
- Fixed unclear RPC logging. Updated the log message to clearly indicate when a password sets the next run time and is not doing a change attempt.
- Improved the placement of Secret Server user admin and role links. They are now on the top level in All Settings under the category header.
- Improved the UI for SSH cipher pages.

Secret Server 11.7.000002 Release Notes

Release Date (On-premises): May 15, 2024

> **Critical security release**—We recommend all Secret Server installations to be updated to this release immediately or at your earliest convenience.

This Security Release corrects the encryption key used in identity token generation to prevent third party decryption and modification of the authentication token, with a CVSS score of 7.5, with vector AV:N/AC:H/PR:H/UI:R/S:C/C:H/I:H/A:L.

This vulnerability was identified internally during our investigation of any earlier vulnerability that was resolved in version 11.7.000001.

> This release invalidates currently issues authentication tokens, and will result in all logon sessions being invalidated. Users may need to reauthenticate to Secret Server after applying this update.

Delinea Platform and Secret Server Cloud

Delinea Platform and Secret Server Cloud have been patched and are no longer vulnerable.

Step Upgrade Process

- A Step Upgrade is required from versions prior to 11.5.2 (11.5.000002) before you can upgrade to 11.7.2 (11.7.000002).
- The automatic downloads in the product will get the right versions for the step upgrade and then allow the 11.7.000002 upgrade.
- If offline and using the file upload method, versions prior to 11.5.2 will get an error message saying, "Integrity Check failed - Security Catalog is signed by thumbprint that is not specifically trusted." The remedy is to first upgrade to 11.5.000002 (or 11.5.000003) and then upgrade to 11.7.000002.

For instructions on upgrading in general, go to "Upgrading Secret Server" on page 125

Secret Server 11.7.000001 Release Notes

Critical security release–We require all Secret Server installations to be updated to this release immediately or at your earliest convenience.

Release Date (On-premises): April 13, 2024

Security Update

We became aware of a critical vulnerability in the SOAP API which could allow an attacker to bypass authentication. The REST API was not impacted.

This update addresses the above security vulnerability and impacts all versions of Secret Server. Hashes the for upgrade have been updated for this change.

Details are available on the Delinea Trust Center. Please register and subscribe to get future updates directly to your inbox.

The direct link to the topic is Secret Server Vulnerability.

Remediation

The direct link to the topic is Secret Server Vulnerability.

Remediation

If your Secret Server instance is exposed to the public internet, you are at significant risk and you should perform these steps immediately.

1. Use the Remediation Guide to modify the Secret Server implementation to mitigate the vulnerability
2. As a precautionary measure rotate your passwords often until mitigation is in place.
3. Use the Remediation Guide to examine audit histories for any evidence of exploitation
4. As soon as the patch is available, patch all systems.

The support team is available to guide your team through these steps and answer any questions from you or your team.

Delinea Platform and Secret Server Cloud

Delinea Platform and Secret Server Cloud have been patched and are no longer vulnerable.

Step Upgrade Process

- A Step Upgrade is required from versions prior to 11.5.2 (11.5.000002) before you can upgrade to 11.7.1 (11.7.000001).
- The automatic downloads in the product will get the right versions for the step upgrade and then allow the 11.7.000001 upgrade.
- If offline and using the file upload method, versions prior to 11.5.2 will get an error message saying, "Integrity

Check failed - Security Catalog is signed by thumbprint that is not specifically trusted." The remedy is to first upgrade to 11.5.000002 (or 11.5.000003) and then do the upgrade to 11.7.000001.

For instructions on upgrading in general, go to "Upgrading" on page 124

If You Cannot Upgrade to 11.7.1

If you are on an older version of Secret Server and you cannot upgrade to the latest version, please contact our support team for assistance and guidance.

- Prevented Thycotic One sync from syncg Platform native users. This allows Platform native users to log in the rare situation they synced with Thycotic One. Then the administrator clears the system Platform User mappings.

Secret Server 11.7.000000 GA Release Notes

On-premises: March 27, 2024

For convenience, this release note also contains items from the 11.7 EA release (11.6.000043), which constitutes most of the changes.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.24.0

Protocol Handler: 6.0.3.27

With this version, protocol handler has received changes to core internal functionality that prevents automatically updating to version 6.0.3.27 from a prior version. In environments with protocol handler automatic update enabled, the protocol handler will automatically update to version 6.0.3.26. To use the latest functionality and fixes of protocol handler, you must redeploy or install version 6.0.3.27 to end-user machines. Following that update, the automatic update will continue to work as before.

Step Upgrade Required (11.5.2). Versions prior to 11.5.2 need to first upgrade to 11.5.2. The automatic downloads in the product will get the right versions for the step upgrade and then allow the 11.6.x upgrade. But if offline and using the file upload method, versions prior to 11.5.2 will get an error message saying, "Integrity Check failed - Security Catalog is signed by thumbprint that is not specifically trusted." The remedy is to first upgrade to 11.5.2 and then do the upgrade to 11.6.x.

For instructions on upgrading in general, go to "Upgrading" on page 124.

New Features

QuantumLock

Secret Server's *QuantumLock* is a feature that provides an additional security layer by protecting secret data using asymmetric encryption (a public/private key pair) where the private key is a human-generated password. This

feature is independent of regular permissions, Secret Server login access, or physical access to the machine running Secret Server.

A shortcut way of thinking about QuantumLocks is as an extra password for secrets that is held by a set group of users. In addition, both the password and the group of users are reusable for other secrets. In addition, QuatumLocks future-proof our digital security infrastructure against the advancing capabilities of quantum computing.

QuantumLock is an upgrade of the earlier doublelock feature. Besides the name change, the difference is QuantumLock offers the option to use a quantum-safe algorithm for encapsulation to protect the private key, specifically CRYSTALS Kyber-1024, which is designed to counter the potential threat from quantum computers to current encryption methods.

Usability Improvements

Left Navigation Panel

With this release, we have made several improvements to the left navigation sub-menu to provide a better user experience. Some of the most common configuration settings have been moved to the top of the menu. For more information about the latest changes, please see "Main Navigation Drawers" on page 162.

Improved Search

The main search is greatly improved. It now includes content search results as well as users. A status bar shows the full folder path for long secret-folder paths. Highlighted search categories include:

- Content
- Favorites
- Folders
- Secrets
- Users

Form Usability Improvements

Overall form information density is improved—forms have less white space between rows.

Configuration Redesign

Our configuration redesign has been available as a preview feature for several versions, and is now fully released.

More content from standalone configuration pages is available directly within the configuration page, as well as section, setting and value search, and is available in Settings > Search configuration

Enhancements

New for GA:

- Enhancement: Improved performance of Secret Search when searching within a folder.
- Enhancement: Custom proxied launchers can now be mapped to a secret's list fields which behaves as an allow-list restriction. This is only allowed without the launcher's "Additional Field" enabled. Only one list field can be used in the mapping.
- Enhancement: Applied several SSH Proxy optimizations to increase performance and throughput.
- Enhancement: Added a warning because WPF allows launching to a freeform <user input> url, which is not recommended.
- Enhancement: Users with "Force Check-in" Permission can now always force secret check in, including the secrets that failed a password change on check-in. Previously, that required "Force check-in" and owner permissions on the secret.
- Enhancement: Password fields on a secret view now indicate if the secret has a password that is non-compliant.
- Enhancement: Added two new password requirement rules that enforce the amount of character repetition allowed in passwords.

Carried over from EA:

- Enhancement: Numerous grids were updated to used the enabled/disabled chip pattern.
- Enhancement: The secret dependencies page was updated. In addition to improved usability, you can filter by groups and run bulk operations across those groups.
- Enhancement: The Dependency Changers page was updated
- Enhancement: The user management user-list grid was updated. It now allows you to perform a bulk action on all users, not just the visibly loaded ones.
- Enhancement: The group list grid was updated.
- Enhancement: Discovery analysis now includes scan results since last run, which is useful for seeing what has succeeded or had an error since the last run.
- Enhancement: A discovery computer-scan results tab was added that allows for searching local computer-scan successes and failures.
- Enhancement: Session recording agent pages were updated.
- Enhancement: ticket system pages were updated.
- Enhancement: Launching a secret now users a dialog flow. This allows launching without having to leave the grid. For example, you can launch and checkout a secret in the launcher flow.
- Enhancement: Diagnostics pages were updated.
- Enhancement: Bulk actions in grids now shows bulk actions at the top of the grid.
- Enhancement: Added search text filter to launcher grid.
- Enhancement: Simplified legacy pages' left navigation design. Removed obsolete setup left navigation menu.
- Enhancement: Secret Server's disaster recovery can now replicate teams, sites, metadata, and secret history.
- Enhancement: Platform users can now use step-up MFA to validate their identity when resetting QuantumLock passwords.

- Enhancement: Added a new setting to the ticket system configuration to optionally avoid prompting for a comment when "Comment not required" is configured.
- Enhancement: Added support for near-real-time processing of Platform user and group updates in Secret Server.
- Enhancement: Added a link to configuration audits on the Remote Password Changing page.
- Enhancement: Added a running log to disaster recovery so progress and duration per table can be tracked during replication.
- Enhancement: Added an event subscription called "Disaster Recovery Replication Success."
- Enhancement: Added auditing of password change schedules.
- Enhancement: Folders in favorites quick access are now filtered when searching.
- Enhancement: Improved HSM cryptography by adding support for AES 256 encryption. This ensures that all keys protecting the secret key will be at the same strength for organizations requiring this level of encryption.
- Enhancement: If an Azure Active Directory configuration in directory services becomes corrupt, you can now view and update the credentials to fix it.
- Enhancement: improved internal security checking around launchers.
- Enhancement: Improved SSH proxy block-command handling in VIM.
- Enhancement: Launching a secret now opens in a dialog allowing launch to occur without leaving the grid or current page. Restricted actions like checkout can be performed in the dialog.
- Enhancement: On the Proxying Configuration page, you can now automatically generate new SSH proxy host keys.
- Enhancement: Platform configuration settings were added to disaster recovery.
- Enhancement: Secret search performance improvements. The secret grid now only requests extended fields that are showing. When column selections are updated, a new request is made only if the extended field choices have changes.
- Enhancement: Secrets grid modal on the Secret Erase Requests search page now auto-scrolls.
- Enhancement: The login policy now supports line breaks.
- Enhancement: The secret search API now has a comma-delimited filter parameter for template IDs, which allows searching beyond IIS URL limits compared to the existing array version. Both are still available.
- Enhancement: The user profile allows for date and time format setting.
- Enhancement: Updated the toast message displayed when saving user preferences to accommodate screen readers.
- Enhancement: Users are no longer redirected from the licensing page.
- Enhancement: When Secret Server Cloud is Platform integrated, there is now an "Add from External Directory" option in secret sharing that allows searching directory sources from Platform to add users or groups.
- Enhancement: Added new optional parameter "nobus=true" to the healthcheck endpoint. This allows a faster response in situations where no lookup of the bus status is required.

- Enhancement: Adjusted the password compliance validation job to process more secrets on each run.
- Enhancement: Discovery port scanner now aborts if elapsed time expires prior to windows TCP handshake. Discovery port scanner will now also log a helpful message if the windows TCP stack aborts due to reaching the windows internal max syn retry count.
- Enhancement: The schedule pipeline task in event pipeline policies now supports using a variable for the schedule delay input.
- Enhancement: Unlimited Admin can now check in checked out secrets by other users.
- Enhancement: Added a Computer Scan Results tab to discovery.
- Enhancement: Added a new option to the Distributed Engine page for configuring "pending engines" that allows a pending engine to be assigned to a site without activation.
- Enhancement: Added a note to audits when the system disables a Secret Server user.
- Enhancement: Added an SDK link.
- Enhancement: Added the table tbTerminalConnectionHistory to the list of tables that is handled by the database cleanup consumer. It will periodically delete any records over a certain age, which can be customized by the user.
- Enhancement: Custom proxied launchers can now be mapped to a secret's list fields which behaves as an allow list restriction. This is only allowed without the launcher's "Additional Field" enabled. Only one list field can be used in the mapping.

Bug Fixes

New for GA:

- Fixed an issue when a syslog server is configured to use a DE and is having connection issues, it can trigger a restart of the DE, interrupting proxy sessions. Now, the syslog circuit breaker does not trigger a restart of the DE.
- Fixed issue with SecureCRT failing to connect to terminal with public key and 2FA on.
- Adjusted the endpoint `/api/v1/secret-access-requests` for better performance.
- Fixed an issue where going to Platform groups and removing (disabling) a group and then searching Platform and re-adding that group made a duplicate instead of enabling the existing group. Additionally, Platform group synchronization now ignores all disabled groups when making membership changes.
- Fixed an issue where a DE unhandled exception disconnected all SSH Proxy users.
- Fixed an issue where if a group has been imported from Platform from an AD source and was then added into directory synchronization as an AD group, it re-used that Platform group rather than creating a new group.
- Fixed an issue where the launcher icon could show when launchers were not allowed.
- Fixed Start Date and Queue Date on "pipeline activity" when viewing an individual run.
- Reduced the frequency of pre-audit validation errors.
- Improved the handling of duplicate platform permissions and added a delta to clean up existing duplicates.
- Fixed an issue where in some configurations secrets could not be created.

- Fixed back-end errors when publishing audits to the audit service.
- Fixed an issue where sorting did not work in card view.
- Explicitly mentioned the Integrated windows authentication requirement on the Database page.
- Fixed an issue where there was no option to select "none" as a default role in Platform/Configuration preview.
- Fixed an issue where encryption-related fields, which are not replicated, were not being populated on replicated sites. These fields were not populated when sites are first saved to the replica.
- Made UI adjustments to ensure that extended fields should be properly requested when column preferences is setup or when using the defaults with extended fields.
- Fixed an issue where the expanded state of the left navigation pane was not preserved correctly.

Carried over from EA:

- Fixed an issue where a distributed engine would not start when proxying was enabled.
- Fixed an issue where the "Add Scanner" Button was not displayed on the Discovery Scanner page.
- Updated installer EULA.
- Fixed an issue where terminating all sessions except the current one would log the user out and report an error.
- Fixed the secret permissions API to handle an edge case that could incorrectly return null when the userid filter parameter was specified.
- Fixed UI to only call secret-detail RPC API when RPC is enabled.
- Fixed the secret-key re-encryptor with a multithreaded lock to prevent an IndexOutOfRangeException while using an HSM.
- Fixed an issue where users checking out a secret with a failed remote password change would potentially see a loading icon indefinitely.
- Fixed an issue where users were redirected to the Secret Server start page instead of the activation center page when using the offline method for license activation.
- Fixed an issue where pre-checkout created an extra pipeline policy activity entry that stayed in a pending state.
- Fixed an issue where "Administer Secret Templates" permission was erroneously required to add a dependency to a secret. "View Secret Templates" is now sufficient.
- Fixed a DR issue where deleting items connected to MetadataItemData orphans it.
- Fixed a session recording issue where a launch would show two records.
- Fixed an issue where Secret Server On Premise launchers list duplicates some items and other launchers do not show up.
- Fixed an issue so the SSH Proxy's processing delay is now respected and defaults to 0, no delay.
- Fixed permission issues with the "View Session Recording" and "View Session Monitoring" roles.
- Fixed an issue where maximum consecutive character rules for passwords did not work and were enforced as expected in the password field.
- Fixed an issue where the group members page was incorrectly showing a maximum of 59 users.

- Fixed a duplicate personal group name when a user got deleted and re-added during disaster recovery replication.
- Fixed when roles were assigned, sometimes no audit record was made.
- Added a warning added because WPF allows launching to a free-form <user input> url, which is not recommended.
- Added "view all folders" link that appears when folders are filtered in a pin view.
- Added a download button for session recording to Secret Server. The change does not appear for vault sessions in Platform.
- Added aria labels to the notification bell to support screen readers.
- Added new REST API patch method to controller which calls pre-existing latestversion.txt processing code.
- Added protocol handler step-up upgrade. Protocol handler will not try to upgrade versions 6.0.3.26 to newer versions as they must be updated manually. Released new 6.0.3.27 version which will be able to upgrade to future versions.
- Adjusted license tracking for session-recording-enabled secrets so that secrets that have no launchers are excluded.
- Adjusted organization of some administrative menu items in the configuration preview.
- Adjusted permissions on Session Monitoring page so that users with "View Own Session Recordings" permission will only see their own recordings.
- Adjusted the display of administrative items from Platform to avoid perceived duplication.
- Adjusted the log level downward for certain engine messages for syslog to avoid overloading the engine log table.
- Applied a more reasonable default SQL timeout.
- Clarified explanatory information on the Secret Import page to highlight that file fields are ignored.
- Converted dependency template management section to new UI.
- Converted Initial User page to the new UI.
- Corrected an issue where the Distributed Engine page did not respect the "Deleted" filter.
- Disabled the legacy bookmarklet pages.
- Disaster recovery now migrates teams.
- Fixed a client-side error on the Secret Settings page when viewed from Platform.
- Fixed a display issue on the IP Address restrictions page.
- Fixed a missing localization-key issue.
- Fixed a visual bug on secret templates so the password type dropdown no longer appears as "None" if a password type has been set.
- Fixed an edge case that could result in duplicate disabled usernames, possibly causing DR conflicts.
- Fixed an error that could occur on the Advanced Session Recording page.

- Fixed an HTML-encoded document link in discovery scanner.
- Fixed an issue an erroneous warning popup appeared saying a distributed engine is required for Active Directory when the SSC cloud instance has "Azure AD Domain" as the only domain.
- Fixed an issue on the Admin Roles page where the edit button for role permissions was mistakenly requiring "Administer Role Assignment" instead of "Administer Role Permission."
- Fixed an issue that could cause an incorrect error message to display when using the SQL report editor.
- Fixed an issue that could cause the secret picker to display with a horizontal scroll bar.
- Fixed an issue when searching in Secret Share with the "Add from External Directory" option with results of more than 2100 groups would throw an error.
- Fixed an issue where a proper validation message may not display when trying to give a duplicate name to a group.
- Fixed an issue where a secret erase request could no longer be canceled.
- Fixed an issue where banner text referenced only "engine," which was potentially confusing. It now mentions "distributed engine" explicitly.
- Fixed an issue where created hooks would not display on the secret.
- Fixed an issue where enabling RPC on a template through the API could impair the template's functionality.
- Fixed an issue where existing linked groups under the Platform Integration area on the Groups tab would not load.
- Fixed an issue where if a non-local site was used to send syslog to the syslog server any failure was queued back into the database (tbsyslogfailedmessage) and resent indefinitely. This has been resolved. Additionally, we implemented a syslog circuit-breaker system if a non-local site is used to prevent flooding the message queues with syslog messages when failure is expected.
- Fixed an issue where localization load requests would wait indefinitely in some cases.
- Fixed an issue where pinned folders would not be removed when the corresponding folder was deleted.
- Fixed an issue where Platform synchronization was running too frequently in some cases.
- Fixed an issue where renaming or copying the "Oracle Account (Template Ver 2)" secret template caused password changes to fail.
- Fixed an issue where Resilient Secrets (DR) sent secret field launchers across the wire for every replication.
- Fixed an issue where selecting Generate New SSH Key on a secret would not generate a new SSH key.
- Fixed an issue where sorting the launchers list by name could display duplicates.
- Fixed an issue where the checkout screen could briefly show while a secret is loading.
- Fixed an issue where the child launcher type was not always visible on the new custom launcher page.
- Fixed an issue where the Everybody group from Platform would not match up properly with the Everybody group from Platform User sync. Corrected the display name of the Platform "Everybody" group.
- Fixed an issue where the light mode collapsed toolbar showed the dark mode logo.
- Fixed an issue where the notification bell could show when there were no notifications.

- Fixed an issue where the Preserve SSH Client Process setting did not correctly display as checked.
- Fixed an issue where the SSH custom cipher was not applied when missing a value from the section.
- Fixed an issue where the synchronized groups displayed could sometimes return all the groups from the domain.
- Fixed an issue where the web launcher would not respect the mapped URL field when multiple URL fields existed on the secret.
- Fixed an issue where unnecessary audits could be written. Fixed an issue where DR Secret Server instances were ignoring licensing updates from Cloud Manager.
- Fixed an issue where upgrade banner was always showing when auto-update was off. Now shows only if at least one engine is lower version than latest.
- Fixed an issue where users could click New Secret multiple times when also uploading files.
- Fixed and incorrect launcher edit field description.
- Fixed buttons that should be grayed out. Run RPC Now can no longer be run when RPC is disabled. Run heartbeat Now can no longer be run when heartbeat is disabled.
- Fixed dark mode IBM password tooltips and banner color-contrast issues.
- Fixed edge case bug if SSH Block Listing causes duplicate sessions that break SSH Proxy.
- Fixed error that could occur when creating a new folder with the folder panel minimized.
- Fixed inconsistent logs between source and replica on partial success. Fatal error is now persisted across the wire so the replica is aware that the source had a fatal error
- Fixed incorrect logging error in AuthenticateWithAdConsumer.
- Fixed issue in directory sync where a search result with an attribute containing an empty list could cause an error.
- Fixed issue where the upper right search bar would not always switch to the selected secret when a selected secret was on a tab other than the General tab.
- Fixed issue with a test script modal where reopening the modal would show the selected secret's ID instead of its name.
- Fixed issue with folder permission editing when updating a path directly.
- Fixed link to dependency templates on the Secret Dependency tab.
- Fixed logic error where the RAS flag was not being referenced before deciding to delete the database entry that reflected additional users.
- Fixed long secret-template names to wrap better in folder edit.
- Fixed missing option. System group in Secret Server Cloud can now have metadata deleted.
- Fixed Platform permissions cached on Secret Server to replicate so they will be respected on a replica instance.
- Fixed query for obtaining services for a directory account in discovery. Fixed check on discovery source name when creating an empty discovery source.

- Fixed secret policies not showing as deleted after deleting a secret. Secret names on the RPC tab of a secret policy will now include "Inactive" if a secret is not active.
- Fixed text alignment. Left aligned the comment text on the MFA security view. The icon and button remain centered.
- Fixed the link to the subscription page from the banner.
- Fixed the REST API token endpoint path. The documentation generator, in removing the "api" string from the beginning of all routes, was also removing embedded occurrences. It now removes it only from the start of the route strings.
- Fixed the secrets grid on the Secret Erase Request Approval page (in a modal opened via a link button) that was obscured in dark mode and nearly indistinguishable in light mode. This is now an inline grid with auto-scroll.
- Fixed visual bug when removing current user's folder owner permissions.
- Folders in "Shared with Me" Quick Access menu are now filtered when searching.
- If a user's encrypted TOTP reset Guid gets corrupted, an administrator is now able to reset their TOTP.
- Improved error handling on the OpenId Configuration page.
- Improved the UI on the Collections Management page for advanced session recording agents.
- In the prior upgrade file set for 11.6.3, fixed an issue with SQL Delta 11.5.000006. Removed a SQL hint on the SQL index that was incompatible with non-Enterprise editions prior to SQL Server 2016 SP1 due to a compatibility issue with data compression. The incompatible hint was not necessary, so the delta was updated. Hashes for upgrade were updated for this change.
- Legacy RPC admin page removed.
- Legacy user and group management aspx pages removed.
- Limited Mode now goes to the correct link in SSC.
- Made performance improvements for the "What Secret Permissions Exist?" report.
- Prevented Thycotic One sync from syncg Platform native users. This allows Platform native users to log in in the rare situation they synced with Thycotic One. Then the administrator clears the system Platform User mappings.
- Queries executed in the chart and SQL editor for custom reports will now take the Use Database Paging setting into account so that the result is the same as if the query was being saved as a report.
- Removed legacy ASPX pages for secret templates.
- Removed link for managing licenses from the Cloud Subscriptions page.
- Secret Server was updated to use the same player for session recordings as platform.
- Set the GET SDK Client Account, SDK Client Audit, and SDK Client Rule API calls to set the operator parameter to 1 if it is not supplied by the caller when a User ID filter is specified.
- Switching pinned folders now resets the text search.
- Updated auditing for users modifying allowed cipher suite algorithms.
- Updated diagnostics page and licensing expiration checks to correctly handle non-US date patterns.
- Updated event subscription and workflow grids.

- Updated password requirement audits to correctly audit missed fields.
- Updated the action-handler secret-launch dialog layout to reflect design changes.
- Updated the Cloud Subscription page to the new UI.
- Updated the Dependency Changes List page to the new UI.
- Updated the Diagnostics page to the new UI.
- Updated the display for secret locked pages to address a wrapping issue with DoubleLock.
- Updated the distributed engine log UI updated. It now remembers your last selected site, system log grid UI updated, and the last selected log level.
- Updated the EventDetails token within Event Subscriptions to correctly capture secret comments.
- Updated the logout.aspx page to avoid errors being generated in rare cases when executing the SAML SLO flow.
- Updated the ticket system list page to the new UI.
- Updated user preferences page for better accessibility.
- web.config now allows explicit definition of allowed HTTP verbs.
- Addressed an issue where discovery rules would not correctly display the selected secret template or password type.
- Adjusted discovery scanning to minimize potential SQL deadlocks during the scanning process.
- Adjusted Secret Server and distributed engine to support 3.x versions of SAP .NET Connector.
- Converted HSM to a new UI page with a new PKCS11 API type. This new option enables you to protect your MEK and secret keys with an AES 256 key, bringing the strength of all keys to AES 256. After setting up PKCS#11 with your HSM vendor, you use the vendor cryptoki library (dll), token label and user pin to integrate with Secret Server. NOTE: You will need to disable the HSM first, to switch to the new PKCS11 API type. See the Hardware Security Module for more details.
- Enable Audit Integration on the Platform Configuration page can now be turned on.
- Extended timeout for some indexing steps for customers with over one million secrets.
- Fixed an issue that caused folder permissions to not update under specific circumstances.
- Fixed an issue where long column names did not wrap in the column selector.
- Fixed an issue where searching for a secret name using a substring within a single word would not always return results.
- Fixed an issue where the Dashboard Overview tab was not selected by default.
- Fixed an issue where the main search did not return content and updated the search design.
- Fixed an issue where the New Folder button would be incorrectly hidden in certain situations when displayed from Platform.
- Fixed an issue where the system log filter preference had an error when all was selected as the last used filter.

- Fixed an issue where the folder tree disappeared when there were more than 1,000 folders accessed and UAM was enabled.
- Fixed an issue with the discovery splash image margin.
- Fixed bug where changing the client ID did not update unless the client secret was updated as well.
- Fixed display issue for Secret edit modal on Discovery scope page.
- Fixed issue with QuantumLock Assign Users grid not displaying correctly after editing then canceling.
- Fixed the folder audit download to show the correct title.
- Improved exception logging for certain scenarios related to launching.
- License requirement message for secret policies updated to Pro Edition or higher.
- Removed no-longer-used bookmarklet login pages.
- Updated API documentation for updating team membership.
- Updated the Secret Import to handle a trailing whitespace in the folder path to prevent bug where created the child folder at the root level.
- Updated the ticket system detail page to the modern UI framework.

Secret Server 11.7.000000 EA Release Notes

On-premises: February 28, 2024

> The 11.7.000000 EA (Early Access) release number is officially 11.6.000043. The official release number will become 11.7.x when the GA (General Availability) release becomes available.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.23.0

Protocol Handler: 6.0.3.27

> With this version, protocol handler has received changes to core internal functionality that prevents automatically updating to version 6.0.3.27 from a prior version. In environments with protocol handler automatic update enabled, the protocol handler will automatically update to version 6.0.3.26. To use the latest functionality and fixes of protocol handler, you must redeploy or install version 6.0.3.27 to end-user machines. Following that update, the automatic update will continue to work as before.

> Step Upgrade Required (11.5.2). Versions prior to 11.5.2 need to first upgrade to 11.5.2. The automatic downloads in the product will get the right versions for the step upgrade and then allow the 11.6.x upgrade. But if offline and using the file upload method, versions prior to 11.5.2 will get an error message saying, "Integrity Check failed - Security Catalog is signed by thumbprint that is not specifically trusted." The remedy is to first upgrade to 11.5.2 and then do the upgrade to 11.6.x.

> For instructions on upgrading in general, go to "Upgrading" on page 124.

New Features

QuantumLock

Secret Server's *QuantumLock* is a feature that provides an additional security layer by protecting secret data using asymmetric encryption (a public/private key pair) where the private key is a human-generated password. This feature is independent of regular permissions, Secret Server login access, or physical access to the machine running Secret Server.

A shortcut way of thinking about QuantumLocks is as an extra password for secrets that is held by a set group of users. In addition, both the password and the group of users are reusable for other secrets. In addition, QuatumLocks future-proof our digital security infrastructure against the advancing capabilities of quantum computing.

QuantumLock is an upgrade of the earlier doublelock feature. Besides the name change, the difference is QuantumLock offers the option to use a quantum-safe algorithm for encapsulation to protect the private key, specifically CRYSTALS Kyber-1024, which is designed to counter the potential threat from quantum computers to current encryption methods.

Usability Improvements

Left Navigation Panel

With this release, we have made several improvements to the left navigation sub-menu to provide a better user experience. Some of the most common configuration settings have been moved to the top of the menu. For more information about the latest changes, please see "Main Navigation Drawers" on page 162.

Improved Search

The main search is greatly improved. It now includes content search results as well as users. A status bar shows the full folder path for long secret-folder paths. Highlighted search categories include:

- Content
- Favorites
- Folders
- Secrets
- Users

Form Usability Improvements

Overall form information density is improved—forms have less white space between rows.

Configuration Redesign

Our configuration redesign has been available as a preview feature for several versions, and is now fully released.

More content from standalone configuration pages is available directly within the configuration page, as well as section, setting and value search, and is available in Settings > Search configuration

Enhancements

- Enhancement: Numerous grids were updated to used the enabled/disabled chip pattern.
- Enhancement: The secret dependencies page was updated. In addition to improved usability, you can filter by groups and run bulk operations across those groups.
- Enhancement: The Dependency Changers page was updated
- Enhancement: The user management user-list grid was updated. It now allows you to perform a bulk action on all users, not just the visibly loaded ones.
- Enhancement: The group list grid was updated.
- Enhancement: Discovery analysis now includes scan results since last run, which is useful for seeing what has succeeded or had an error since the last run.
- Enhancement: A discovery computer-scan results tab was added that allows for searching local computer-scan successes and failures.
- Enhancement: Session recording agent pages were updated.
- Enhancement: ticket system pages were updated.
- Enhancement: Launching a secret now users a dialog flow. This allows launching without having to leave the grid. For example, you can launch and checkout a secret in the launcher flow.
- Enhancement: Diagnostics pages were updated.
- Enhancement: Bulk actions in grids now shows bulk actions at the top of the grid.
- Enhancement: Added search text filter to launcher grid.
- Enhancement: Simplified legacy pages' left navigation design. Removed obsolete setup left navigation menu.
- Enhancement: Secret Server's disaster recovery can now replicate teams, sites, metadata, and secret history.
- Enhancement: Platform users can now use step-up MFA to validate their identity when resetting QuantumLock passwords.
- Enhancement: Added a new setting to the ticket system configuration to optionally avoid prompting for a comment when "Comment not required" is configured.
- Enhancement: Added support for near-real-time processing of Platform user and group updates in Secret Server.
- Enhancement: Added a link to configuration audits on the Remote Password Changing page.
- Enhancement: Added a running log to disaster recovery so progress and duration per table can be tracked during replication.
- Enhancement: Added an event subscription called "Disaster Recovery Replication Success."
- Enhancement: Added auditing of password change schedules.
- Enhancement: Folders in favorites quick access are now filtered when searching.
- Enhancement: Improved HSM cryptography by adding support for AES 256 encryption. This ensures that all keys protecting the secret key will be at the same strength for organizations requiring this level of encryption.

- Enhancement: If an Azure Active Directory configuration in directory services becomes corrupt, you can now view and update the credentials to fix it.
- Enhancement: improved internal security checking around launchers.
- Enhancement: Improved SSH proxy block-command handling in VIM.
- Enhancement: Launching a secret now opens in a dialog allowing launch to occur without leaving the grid or current page. Restricted actions like checkout can be performed in the dialog.
- Enhancement: On the Proxying Configuration page, you can now automatically generate new SSH proxy host keys.
- Enhancement: Platform configuration settings were added to disaster recovery.
- Enhancement: Secret search performance improvements. The secret grid now only requests extended fields that are showing. When column selections are updated, a new request is made only if the extended field choices have changes.
- Enhancement: Secrets grid modal on the Secret Erase Requests search page now auto-scrolls.
- Enhancement: The login policy now supports line breaks.
- Enhancement: The secret search API now has a comma-delimited filter parameter for template IDs, which allows searching beyond IIS URL limits compared to the existing array version. Both are still available.
- Enhancement: The user profile allows for date and time format setting.
- Enhancement: Updated the toast message displayed when saving user preferences to accommodate screen readers.
- Enhancement: Users are no longer redirected from the licensing page.
- Enhancement: When Secret Server Cloud is Platform integrated, there is now an "Add from External Directory" option in secret sharing that allows searching directory sources from Platform to add users or groups.
- Enhancement: Added new optional parameter "nobus=true" to the healthcheck endpoint. This allows a faster response in situations where no lookup of the bus status is required.
- Enhancement: Adjusted the password compliance validation job to process more secrets on each run.
- Enhancement: Discovery port scanner now aborts if elapsed time expires prior to windows TCP handshake. Discovery port scanner will now also log a helpful message if the windows TCP stack aborts due to reaching the windows internal max syn retry count.
- Enhancement: The schedule pipeline task in event pipeline policies now supports using a variable for the schedule delay input.
- Enhancement: Unlimited Admin can now check in checked out secrets by other users.
- Enhancement: Added a Computer Scan Results tab to discovery.
- Enhancement: Added a new option to the Distributed Engine page for configuring "pending engines" that allows a pending engine to be assigned to a site without activation.
- Enhancement: Added a note to audits when the system disables a Secret Server user.
- Enhancement: Added an SDK link.

- Enhancement: Added the table tbTerminalConnectionHistory to the list of tables that is handled by the database cleanup consumer. It will periodically delete any records over a certain age, which can be customized by the user.
- Enhancement: Custom proxied launchers can now be mapped to a secret's list fields which behaves as an allow list restriction. This is only allowed without the launcher's "Additional Field" enabled. Only one list field can be used in the mapping.

Bug Fixes

- Fixed an issue where a distributed engine would not start when proxying was enabled.
- Fixed an issue where the "Add Scanner" Button was not displayed on the Discovery Scanner page.
- Updated installer EULA.
- Fixed an issue where terminating all sessions except the current one would log the user out and report an error.
- Fixed the secret permissions API to handle an edge case that could incorrectly return null when the userid filter parameter was specified.
- Fixed UI to only call secret-detail RPC API when RPC is enabled.
- Fixed the secret-key re-encryptor with a multithreaded lock to prevent an IndexOutOfRangeException while using an HSM.
- Fixed an issue where users checking out a secret with a failed remote password change would potentially see a loading icon indefinitely.
- Fixed an issue where users were redirected to the Secret Server start page instead of the activation center page when using the offline method for license activation.
- Fixed an issue where pre-checkout created an extra pipeline policy activity entry that stayed in a pending state.
- Fixed an issue where "Administer Secret Templates" permission was erroneously required to add a dependency to a secret. "View Secret Templates" is now sufficient.
- Fixed a DR issue where deleting items connected to MetadataItemData orphans it.
- Fixed a session recording issue where a launch would show two records.
- Fixed an issue where Secret Server On Premise launchers list duplicates some items and other launchers do not show up.
- Fixed an issue so the SSH Proxy's processing delay is now respected and defaults to 0, no delay.
- Fixed permission issues with the "View Session Recording" and "View Session Monitoring" roles.
- Fixed an issue where maximum consecutive character rules for passwords did not work and were enforced as expected in the password field.
- Fixed an issue where the group members page was incorrectly showing a maximum of 59 users.
- Fixed a duplicate personal group name when a user got deleted and re-added during disaster recovery replication.
- Fixed when roles were assigned, sometimes no audit record was made.

- Added a warning added because WPF allows launching to a free-form <user input> url, which is not recommended.
- Added "view all folders" link that appears when folders are filtered in a pin view.
- Added a download button for session recording to Secret Server. The change does not appear for vault sessions in Platform.
- Added aria labels to the notification bell to support screen readers.
- Added new REST API patch method to controller which calls pre-existing latestversion.txt processing code.
- Added protocol handler step-up upgrade. Protocol handler will not try to upgrade versions 6.0.3.26 to newer versions as they must be updated manually. Released new 6.0.3.27 version which will be able to upgrade to future versions.
- Adjusted license tracking for session-recording-enabled secrets so that secrets that have no launchers are excluded.
- Adjusted organization of some administrative menu items in the configuration preview.
- Adjusted permissions on Session Monitoring page so that users with "View Own Session Recordings" permission will only see their own recordings.
- Adjusted the display of administrative items from Platform to avoid perceived duplication.
- Adjusted the log level downward for certain engine messages for syslog to avoid overloading the engine log table.
- Applied a more reasonable default SQL timeout.
- Clarified explanatory information on the Secret Import page to highlight that file fields are ignored.
- Converted dependency template management section to new UI.
- Converted Initial User page to the new UI.
- Corrected an issue where the Distributed Engine page did not respect the "Deleted" filter.
- Disabled the legacy bookmarklet pages.
- Disaster recovery now migrates teams.
- Fixed a client-side error on the Secret Settings page when viewed from Platform.
- Fixed a display issue on the IP Address restrictions page.
- Fixed a missing localization-key issue.
- Fixed a visual bug on secret templates so the password type dropdown no longer appears as "None" if a password type has been set.
- Fixed an edge case that could result in duplicate disabled usernames, possibly causing DR conflicts.
- Fixed an error that could occur on the Advanced Session Recording page.
- Fixed an HTML-encoded document link in discovery scanner.
- Fixed an issue an erroneous warning popup appeared saying a distributed engine is required for Active Directory when the SSC cloud instance has "Azure AD Domain" as the only domain.

- Fixed an issue on the Admin Roles page where the edit button for role permissions was mistakenly requiring "Administer Role Assignment" instead of "Administer Role Permission."
- Fixed an issue that could cause an incorrect error message to display when using the SQL report editor.
- Fixed an issue that could cause the secret picker to display with a horizontal scroll bar.
- Fixed an issue when searching in Secret Share with the "Add from External Directory" option with results of more than 2100 groups would throw an error.
- Fixed an issue where a proper validation message may not display when trying to give a duplicate name to a group.
- Fixed an issue where a secret erase request could no longer be canceled.
- Fixed an issue where banner text referenced only "engine," which was potentially confusing. It now mentions "distributed engine" explicitly.
- Fixed an issue where created hooks would not display on the secret.
- Fixed an issue where enabling RPC on a template through the API could impair the template's functionality.
- Fixed an issue where existing linked groups under the Platform Integration area on the Groups tab would not load.
- Fixed an issue where if a non-local site was used to send syslog to the syslog server any failure was queued back into the database (tbsyslogfailedmessage) and resent indefinitely. This has been resolved. Additionally, we implemented a syslog circuit-breaker system if a non-local site is used to prevent flooding the message queues with syslog messages when failure is expected.
- Fixed an issue where localization load requests would await indefinitely in some cases.
- Fixed an issue where pinned folders would not be removed when the corresponding folder was deleted.
- Fixed an issue where Platform synchronization was running too frequently in some cases.
- Fixed an issue where renaming or copying the "Oracle Account (Template Ver 2)" secret template caused password changes to fail.
- Fixed an issue where Resilient Secrets (DR) sent secret field launchers across the wire for every replication.
- Fixed an issue where selecting Generate New SSH Key on a secret would not generate a new SSH key.
- Fixed an issue where sorting the launchers list by name could display duplicates.
- Fixed an issue where the checkout screen could briefly show while a secret is loading.
- Fixed an issue where the child launcher type was not always visible on the new custom launcher page.
- Fixed an issue where the Everybody group from Platform would not match up properly with the Everybody group from Platform User sync. Corrected the display name of the Platform "Everybody" group.
- Fixed an issue where the light mode collapsed toolbar showed the dark mode logo.
- Fixed an issue where the notification bell could show when there were no notifications.
- Fixed an issue where the Preserve SSH Client Process setting did not correctly display as checked.
- Fixed an issue where the SSH custom cipher was not applied when missing a value from the section.

- Fixed an issue where the synchronized groups displayed could sometimes return all the groups from the domain.
- Fixed an issue where the web launcher would not respect the mapped URL field when multiple URL fields existed on the secret.
- Fixed an issue where unnecessary audits could be written. Fixed an issue where DR Secret Server instances were ignoring licensing updates from Cloud Manager.
- Fixed an issue where upgrade banner was always showing when auto-update was off. Now shows only if at least one engine is lower version than latest.
- Fixed an issue where users could click New Secret multiple times when also uploading files.
- Fixed and incorrect launcher edit field description.
- Fixed buttons that should be grayed out. Run RPC Now can no longer be run when RPC is disabled. Run heartbeat Now can no longer be run when heartbeat is disabled.
- Fixed dark mode IBM password tooltips and banner color-contrast issues.
- Fixed edge case bug if SSH Block Listing causes duplicate sessions that break SSH Proxy.
- Fixed error that could occur when creating a new folder with the folder panel minimized.
- Fixed inconsistent logs between source and replica on partial success. Fatal error is now persisted across the wire so the replica is aware that the source had a fatal error
- Fixed incorrect logging error in AuthenticateWithAdConsumer.
- Fixed issue in directory sync where a search result with an attribute containing an empty list could cause an error.
- Fixed issue where the upper right search bar would not always switch to the selected secret when a selected secret was on a tab other than the General tab.
- Fixed issue with a test script modal where reopening the modal would show the selected secret's ID instead of its name.
- Fixed issue with folder permission editing when updating a path directly.
- Fixed link to dependency templates on the Secret Dependency tab.
- Fixed logic error where the RAS flag was not being referenced before deciding to delete the database entry that reflected additional users.
- Fixed long secret-template names to wrap better in folder edit.
- Fixed missing option. System group in Secret Server Cloud can now have metadata deleted.
- Fixed Platform permissions cached on Secret Server to replicate so they will be respected on a replica instance.
- Fixed query for obtaining services for a directory account in discovery. Fixed check on discovery source name when creating an empty discovery source.
- Fixed secret policies not showing as deleted after deleting a secret. Secret names on the RPC tab of a secret policy will now include "Inactive" if a secret is not active.

- Fixed text alignment. Left aligned the comment text on the MFA security view. The icon and button remain centered.
- Fixed the link to the subscription page from the banner.
- Fixed the REST API token endpoint path. The documentation generator, in removing the "api" string from the beginning of all routes, was also removing embedded occurrences. It now removes it only from the start of the route strings.
- Fixed the secrets grid on the Secret Erase Request Approval page (in a modal opened via a link button) that was obscured in dark mode and nearly indistinguishable in light mode. This is now an inline grid with auto-scroll.
- Fixed visual bug when removing current user's folder owner permissions.
- Folders in "Shared with Me" Quick Access menu are now filtered when searching.
- If a user's encrypted TOTP reset Guid gets corrupted, an administrator is now able to reset their TOTP.
- Improved error handling on the OpenId Configuration page.
- Improved the UI on the Collections Management page for advanced session recording agents.
- In the prior upgrade file set for 11.6.3, fixed an issue with SQL Delta 11.5.000006. Removed a SQL hint on the SQL index that was incompatible with non-Enterprise editions prior to SQL Server 2016 SP1 due to a compatibility issue with data compression. The incompatible hint was not necessary, so the delta was updated. Hashes for upgrade were updated for this change.
- Legacy RPC admin page removed.
- Legacy user and group management aspx pages removed.
- Limited Mode now goes to the correct link in SSC.
- Made performance improvements for the "What Secret Permissions Exist?" report.
- Prevented Thycotic One sync from syncg Platform Native users. This allows Platform native users to log in in the rare situation they synced with Thycotic One. Then the administrator clears the system Platform User mappings.
- Queries executed in the chart and SQL editor for custom reports will now take the Use Database Paging setting into account so that the result is the same as if the query was being saved as a report.
- Removed legacy ASPX pages for secret templates.
- Removed link for managing licenses from the Cloud Subscriptions page.
- Secret Server was updated to use the same player for session recordings as platform.
- Set the GET SDK Client Account, SDK Client Audit, and SDK Client Rule API calls to set the operator parameter to 1 if it is not supplied by the caller when a User ID filter is specified.
- Switching pinned folders now resets the text search.
- Updated auditing for users modifying allowed cipher suite algorithms.
- Updated diagnostics page and licensing expiration checks to correctly handle non-US date patterns.
- Updated event subscription and workflow grids.
- Updated password requirement audits to correctly audit missed fields.

- Updated the action-handler secret-launch dialog layout to reflect design changes.
- Updated the Cloud Subscription page to the new UI.
- Updated the Dependency Changes List page to the new UI.
- Updated the Diagnostics page to the new UI.
- Updated the display for secret locked pages to address a wrapping issue with DoubleLock.
- Updated the distributed engine log UI updated. It now remembers your last selected site, system log grid UI updated, and the last selected log level.
- Updated the EventDetails token within Event Subscriptions to correctly capture secret comments.
- Updated the logout.aspx page to avoid errors being generated in rare cases when executing the SAML SLO flow.
- Updated the ticket system list page to the new UI.
- Updated user preferences page for better accessibility.
- web.config now allows explicit definition of allowed HTTP verbs.
- Addressed an issue where discovery rules would not correctly display the selected secret template or password type.
- Adjusted discovery scanning to minimize potential SQL deadlocks during the scanning process.
- Adjusted Secret Server and distributed engine to support 3.x versions of SAP .NET Connector.
- Converted HSM to a new UI page with a new PKCS11 API type. This new option enables you to protect your MEK and secret keys with an AES 256 key, bringing the strength of all keys to AES 256. After setting up PKCS#11 with your HSM vendor, you use the vendor cryptoki library (dll), token label and user pin to integrate with Secret Server. NOTE: You will need to disable the HSM first, to switch to the new PKCS11 API type. See the Hardware Security Module for more details.
- Enable Audit Integration on the Platform Configuration page can now be turned on.
- Extended timeout for some indexing steps for customers with over one million secrets.
- Fixed an issue that caused folder permissions to not update under specific circumstances.
- Fixed an issue where long column names did not wrap in the column selector.
- Fixed an issue where searching for a secret name using a substring within a single word would not always return results.
- Fixed an issue where the Dashboard Overview tab was not selected by default.
- Fixed an issue where the main search did not return content and updated the search design.
- Fixed an issue where the New Folder button would be incorrectly hidden in certain situations when displayed from Platform.
- Fixed an issue where the system log filter preference had an error when all was selected as the last used filter.
- Fixed an issue where the folder tree disappeared when there were more than 1,000 folders accessed and UAM was enabled.

- Fixed an issue with the discovery splash image margin.
- Fixed bug where changing the client ID did not update unless the client secret was updated as well.
- Fixed display issue for Secret edit modal on Discovery scope page.
- Fixed issue with QuantumLock Assign Users grid not displaying correctly after editing then canceling.
- Fixed the folder audit download to show the correct title.
- Improved exception logging for certain scenarios related to launching.
- License requirement message for secret policies updated to Pro Edition or higher.
- Removed no-longer-used bookmarklet login pages.
- Updated API documentation for updating team membership.
- Updated the Secret Import to handle a trailing whitespace in the folder path to prevent bug where created the child folder at the root level.
- Updated the ticket system detail page to the modern UI framework.

Secret Server 11.6.000025 GA Release Notes

On-premises: January 24, 2024

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.21.0

Protocol Handler: 6.0.3.27

> With this version, protocol handler has received changes to core internal functionality that prevents automatically updating to version 6.0.3.27 from a prior version. In environments with protocol handler automatic update enabled, the protocol handler will automatically update to version 6.0.3.26. To use the latest functionality and fixes of protocol handler, you must redeploy or install version 6.0.3.27 to end-user machines. Following that update, the automatic update will continue to work as before.

> Step Upgrade Required (11.5.2). Versions prior to 11.5.2 need to first upgrade to 11.5.2. The automatic downloads in the product will get the right versions for the step upgrade and then allow the 11.6.x upgrade. But if offline and using the file upload method, versions prior to 11.5.2 will get an error message saying, "Integrity Check failed - Security Catalog is signed by thumbprint that is not specifically trusted." The remedy is to first upgrade to 11.5.2 and then do the upgrade to 11.6.x.

> For instructions on upgrading in general, go to "Upgrading" on page 124.

New Features

HSM PKCS #11 API Support and AES256 Key Integration

We are pleased to announce a significant enhancement in the Hardware Security Module (HSM) functionality. This enhancement marks the integration of support for the PKCS #11 APIs, achieved through the use of vendor-

provided DLLs. This development not only extends our compatibility with a broader range of cryptographic tokens and HSMs that comply with the PKCS #11 standard but also ensures optimized performance across HSM models and manufacturers.

Moreover, this update provides the capability to leverage Advanced Encryption Standard (AES) 256-bit keys within the HSM, affording security and performance benefits, bringing our HSM integration in line with the majority of cryptographic functions used elsewhere in our products. This underscores our commitment to providing top-tier security features, crucial for safeguarding sensitive data in high-security environments.

The integration of PKCS #11 API support and AES256 not only enhances our security landscape but also offers increased flexibility and compatibility. This update enables users to more effectively use their existing HSM more effectively and provides an expanded array of cryptographic operations options, meeting diverse security needs.

Detailed information on configuring and utilizing the new HSM functions can be found in our updated documentation See "Using Hardware Security Modules " on page 1440 for details.

> You will need to disable the HSM first to switch to the new PKCS11 API type. See "Using Hardware Security Modules " on page 1440 for details.

Enhancements

- Enhancement: Added a link to configuration audits on the Remote Password Changing page.
- Enhancement: Added a running log to disaster recovery so progress and duration per table can be tracked during replication.
- Enhancement: Added an event subscription called "Disaster Recovery Replication Success."
- Enhancement: Added auditing of password change schedules.
- Enhancement: Folders in favorites quick access are now filtered when searching.
- Enhancement: Improved HSM cryptography by adding support for AES 256 encryption. This ensures that all keys protecting the secret key will be at the same strength for organizations requiring this level of encryption.
- Enhancement: If an Azure Active Directory configuration in directory services becomes corrupt, you can now view and update the credentials to fix it.
- Enhancement: improved internal security checking around launchers.
- Enhancement: Improved SSH proxy block-command handling in VIM.
- Enhancement: Launching a secret now opens in a dialog allowing launch to occur without leaving the grid or current page. Restricted actions like checkout can be performed in the dialog.
- Enhancement: On the Proxying Configuration page, you can now automatically generate new SSH proxy host keys.
- Enhancement: Platform configuration settings were added to disaster recovery.
- Enhancement: Secret search performance improvements. The secret grid now only requests extended fields that are showing. When column selections are updated, a new request is made only if the extended field choices have changes.
- Enhancement: Secrets grid modal on the Secret Erase Requests search page now auto-scrolls.

- Enhancement: The login policy now supports line breaks.
- Enhancement: The secret search API now has a comma-delimited filter parameter for template IDs, which allows searching beyond IIS URL limits compared to the existing array version. Both are still available.
- Enhancement: The user profile allows for date and time format setting.
- Enhancement: Updated the toast message displayed when saving user preferences to accommodate screen readers.
- Enhancement: Users are no longer redirected from the licensing page.
- Enhancement: When Secret Server Cloud is Platform integrated, there is now an "Add from External Directory" option in secret sharing that allows searching directory sources from Platform to add users or groups.

Bug Fixes

- Added "view all folders" link that appears when folders are filtered in a pin view.
- Added a download button for session recording to Secret Server. The change does not appear for vault sessions in Platform.
- Added aria labels to the notification bell to support screen readers.
- Added new REST API patch method to controller which calls pre-existing latestversion.txt processing code.
- Added protocol handler step-up upgrade. Protocol handler will not try to upgrade versions 6.0.3.26 to newer versions as they must be updated manually. Released new 6.0.3.27 version which will be able to upgrade to future versions.
- Adjusted license tracking for session-recording-enabled secrets so that secrets that have no launchers are excluded.
- Adjusted organization of some administrative menu items in the configuration preview.
- Adjusted permissions on Session Monitoring page so that users with "View Own Session Recordings" permission will only see their own recordings.
- Adjusted the display of administrative items from Platform to avoid perceived duplication.
- Adjusted the log level downward for certain engine messages for syslog to avoid overloading the engine log table.
- Applied a more reasonable default SQL timeout.
- Clarified explanatory information on the Secret Import page to highlight that file fields are ignored.
- Converted dependency template management section to new UI.
- Converted Initial User page to the new UI.
- Corrected an issue where the Distributed Engine page did not respect the "Deleted" filter.
- Disabled the legacy bookmarklet pages.
- Disaster recovery now migrates teams.
- Fixed a client-side error on the Secret Settings page when viewed from Platform.
- Fixed a display issue on the IP Address restrictions page.

- Fixed a missing localization-key issue.
- Fixed a visual bug on secret templates so the password type dropdown no longer appears as "None" if a password type has been set.
- Fixed an edge case that could result in duplicate disabled usernames, possibly causing DR conflicts.
- Fixed an error that could occur on the Advanced Session Recording page.
- Fixed an HTML-encoded document link in discovery scanner.
- Fixed an issue an erroneous warning popup appeared saying a distributed engine is required for Active Directory when the SSC cloud instance has "Azure AD Domain" as the only domain.
- Fixed an issue on the Admin Roles page where the edit button for role permissions was mistakenly requiring "Administer Role Assignment" instead of "Administer Role Permission."
- Fixed an issue that could cause an incorrect error message to display when using the SQL report editor.
- Fixed an issue that could cause the secret picker to display with a horizontal scroll bar.
- Fixed an issue when searching in Secret Share with the "Add from External Directory" option with results of more than 2100 groups would throw an error.
- Fixed an issue where a proper validation message may not display when trying to give a duplicate name to a group.
- Fixed an issue where a secret erase request could no longer be canceled.
- Fixed an issue where banner text referenced only "engine," which was potentially confusing. It now mentions "distributed engine" explicitly.
- Fixed an issue where created hooks would not display on the secret.
- Fixed an issue where enabling RPC on a template through the API could impair the template's functionality.
- Fixed an issue where existing linked groups under the Platform Integration area on the Groups tab would not load.
- Fixed an issue where if a non-local site was used to send syslog to the syslog server any failure was queued back into the database (tbsyslogfailedmessage) and resent indefinitely. This has been resolved. Additionally, we implemented a syslog circuit-breaker system if a non-local site is used to prevent flooding the message queues with syslog messages when failure is expected.
- Fixed an issue where localization load requests would await indefinitely in some cases.
- Fixed an issue where pinned folders would not be removed when the corresponding folder was deleted.
- Fixed an issue where Platform synchronization was running too frequently in some cases.
- Fixed an issue where renaming or copying the "Oracle Account (Template Ver 2)" secret template caused password changes to fail.
- Fixed an issue where Resilient Secrets (DR) sent secret field launchers across the wire for every replication.
- Fixed an issue where selecting Generate New SSH Key on a secret would not generate a new SSH key.
- Fixed an issue where sorting the launchers list by name could display duplicates.
- Fixed an issue where the checkout screen could briefly show while a secret is loading.

- Fixed an issue where the child launcher type was not always visible on the new custom launcher page.
- Fixed an issue where the Everybody group from Platform would not match up properly with the Everybody group from Platform User sync. Corrected the display name of the Platform "Everybody" group.
- Fixed an issue where the light mode collapsed toolbar showed the dark mode logo.
- Fixed an issue where the notification bell could show when there were no notifications.
- Fixed an issue where the Preserve SSH Client Process setting did not correctly display as checked.
- Fixed an issue where the SSH custom cipher was not applied when missing a value from the section.
- Fixed an issue where the synchronized groups displayed could sometimes return all the groups from the domain.
- Fixed an issue where the web launcher would not respect the mapped URL field when multiple URL fields existed on the secret.
- Fixed an issue where unnecessary audits could be written. Fixed an issue where DR Secret Server instances were ignoring licensing updates from Cloud Manager.
- Fixed an issue where upgrade banner was always showing when auto-update was off. Now shows only if at least one engine is lower version than latest.
- Fixed an issue where users could click New Secret multiple times when also uploading files.
- Fixed and incorrect launcher edit field description.
- Fixed buttons that should be grayed out. Run RPC Now can no longer be run when RPC is disabled. Run heartbeat Now can no longer be run when heartbeat is disabled.
- Fixed dark mode IBM password tooltips and banner color-contrast issues.
- Fixed edge case bug if SSH Block Listing causes duplicate sessions that break SSH Proxy.
- Fixed error that could occur when creating a new folder with the folder panel minimized.
- Fixed inconsistent logs between source and replica on partial success. Fatal error is now persisted across the wire so the replica is aware that the source had a fatal error
- Fixed incorrect logging error in AuthenticateWithAdConsumer.
- Fixed issue in directory sync where a search result with an attribute containing an empty list could cause an error.
- Fixed issue where the upper right search bar would not always switch to the selected secret when a selected secret was on a tab other than the General tab.
- Fixed issue with a test script modal where reopening the modal would show the selected secret's ID instead of its name.
- Fixed issue with folder permission editing when updating a path directly.
- Fixed link to dependency templates on the Secret Dependency tab.
- Fixed logic error where the RAS flag was not being referenced before deciding to delete the database entry that reflected additional users.
- Fixed long secret-template names to wrap better in folder edit.

- Fixed missing option. System group in Secret Server Cloud can now have metadata deleted.
- Fixed Platform permissions cached on Secret Server to replicate so they will be respected on a replica instance.
- Fixed query for obtaining services for a directory account in discovery. Fixed check on discovery source name when creating an empty discovery source.
- Fixed secret policies not showing as deleted after deleting a secret. Secret names on the RPC tab of a secret policy will now include "Inactive" if a secret is not active.
- Fixed text alignment. Left aligned the comment text on the MFA security view. The icon and button remain centered.
- Fixed the link to the subscription page from the banner.
- Fixed the REST API token endpoint path. The documentation generator, in removing the "api" string from the beginning of all routes, was also removing embedded occurrences. It now removes it only from the start of the route strings.
- Fixed the secrets grid on the Secret Erase Request Approval page (in a modal opened via a link button) that was obscured in dark mode and nearly indistinguishable in light mode. This is now an inline grid with auto-scroll.
- Fixed visual bug when removing current user's folder owner permissions.
- Folders in "Shared with Me" Quick Access menu are now filtered when searching.
- If a user's encrypted TOTP reset Guid gets corrupted, an administrator is now able to reset their TOTP.
- Improved error handling on the OpenId Configuration page.
- Improved the UI on the Collections Management page for advanced session recording agents.
- In the prior upgrade file set for 11.6.3, fixed an issue with SQL Delta 11.5.000006. Removed a SQL hint on the SQL index that was incompatible with non-Enterprise editions prior to SQL Server 2016 SP1 due to a compatibility issue with data compression. The incompatible hint was not necessary, so the delta was updated. Hashes for upgrade were updated for this change.
- Legacy RPC admin page removed.
- Legacy user and group management aspx pages removed.
- Limited Mode now goes to the correct link in SSC.
- Made performance improvements for the "What Secret Permissions Exist?" report.
- Prevented Thycotic One sync from syncg Platform Native users. This allows Platform native users to log in in the rare situation they synced with Thycotic One. Then the administrator clears the system Platform User mappings.
- Queries executed in the chart and SQL editor for custom reports will now take the Use Database Paging setting into account so that the result is the same as if the query was being saved as a report.
- Removed legacy ASPX pages for secret templates.
- Removed link for managing licenses from the Cloud Subscriptions page.
- Secret Server was updated to use the same player for session recordings as platform.

- Set the GET SDK Client Account, SDK Client Audit, and SDK Client Rule API calls to set the operator parameter to 1 if it is not supplied by the caller when a User ID filter is specified.
- Switching pinned folders now resets the text search.
- Updated auditing for users modifying allowed cipher suite algorithms.
- Updated diagnostics page and licensing expiration checks to correctly handle non-US date patterns.
- Updated event subscription and workflow grids.
- Updated password requirement audits to correctly audit missed fields.
- Updated the action-handler secret-launch dialog layout to reflect design changes.
- Updated the Cloud Subscription page to the new UI.
- Updated the Dependency Changes List page to the new UI.
- Updated the Diagnostics page to the new UI.
- Updated the display for secret locked pages to address a wrapping issue with DoubleLock.
- Updated the distributed engine log UI updated. It now remembers your last selected site, system log grid UI updated, and the last selected log level.
- Updated the EventDetails token within Event Subscriptions to correctly capture secret comments.
- Updated the logout.aspx page to avoid errors being generated in rare cases when executing the SAML SLO flow.
- Updated the ticket system list page to the new UI.
- Updated user preferences page for better accessibility.
- web.config now allows explicit definition of allowed HTTP verbs.

Secret Server 11.6.000004 Release Notes

On-premises: December 6, 2023

Important security release—we recommend all affected Secret Server On-Premise customers upgrade as soon as possible. This update addresses a security vulnerability recently discovered during internal testing and impacts all versions of Secret Server. A SQL Injection vulnerability was found in the REST API. Hashes for upgrade have been updated for this change. This issue is rated HIGH with a score of 7.2 on the Common Vulnerability Scoring System (CVSS): CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H

This vulnerability has been patched in Secret Server Cloud, so there is no additional update to address it.

The minimum required engine version is 8.3.0.0.

Step Upgrade Required (11.5.2). Versions prior to 11.5.2 need to first upgrade to 11.5.2. The automatic downloads in the product will get the right versions for the step upgrade and then allow the 11.6 upgrade. But if offline and using the file upload method, versions prior to 11.5.2 will get an error message saying, "Integrity Check failed - Security Catalog is signed by thumbprint that is not specifically trusted." The remedy is to first upgrade to 11.5.2 and then do the upgrade to 11.6.4.

For instructions on upgrading in general, go to "Upgrading" on page 124.

Secret Server 11.6.000003 GA Release Notes

For convenience, this document contains the release notes from the 11.6.000000 EA and 11.6.000002 GA releases.

Release Date and Notes

A new Delinea certificate signature has been applied to our binary files so if you are using the old Thycotic Certificate signature, you need to update to the new or you may experience an interruption of service.

We identified and resolved an issue with the latest Secret Server 11.6.000002 for some On-Premise users. Although the issue was intermittent and inconsistent, out of an abundance of caution and a commitment to quality, we deployed Secret Server 11.6.3. Please download Secret Server 11.6.3 at your earliest convenience.

On-premises: September 28, 2023 (September 26, 2023 GA)

This release contains a silent update that fixed an issue with SQL Delta 11.5.000006. We removed a SQL hint on the SQL index that was incompatible with non-Enterprise editions prior to SQL Server 2016 SP1 due to a compatibility issue with data compression. The incompatible hint was not necessary so the delta was updated. Hashes for upgrade have been updated for this change.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.16.0

Protocol Handler: 6.0.3.26

Note: Step Upgrade Required (11.5.2). Versions prior to 11.5.2 will need to first upgrade to 11.5.2. The automatic downloads in the product will get the right versions for the step upgrade and then allow the 11.6 upgrade. But if offline and using the file upload method, versions prior to 11.5.2 will get an error message saying, "Integrity Check failed - Security Catalog is signed by thumbprint that is not specifically trusted." The remedy is to first upgrade to 11.5.2 and then do the upgrade to 11.6.

Feature Enhancements

View Own Session Recordings Permission

A new permission has been added "View Own Session Recordings". With this permission, a user can be restricted to only viewing the recordings that they initiated. If the user with this permission clicks on a recording initiated and owned by another user, they will get an Access Denied window. In addition, the "View Session Recordings" permission has been renamed to "View All Session Recordings".

Session Monitoring Playback Page UI Conversion

The Session Recording playback page has been updated to the new UI, including a new video player with additional controls. The SSH keystroke-only playback page and the video playback page have been merged, and the available elements will be shown. The legacy player is still available as a link from the new video playback page.

> **Note:** The activity graph and download button have not yet been implemented for the new page, although they are available on the legacy page if needed.

Manual Password Change for Checked Out Secrets

Secrets with Change Password on Checkin configured now have the "Change Password Now" functionality available. This will enable the standard functionality of a password change, and the secret will also complete the automatic password change on checking in. This is to allow maintenance and testing of secrets protected in this manner, and a pending password change must be completed before the check-in process is allowed to begin in order to maintain a secure order of operations.

Updated User Selection Interface

Various locations around the product provide a user selection interface to provide the ability to select a user as the target of a particular configuration, such as the permissions, groups, and roles pages. These have been updated throughout the product to provide more data about the users in the list. You may now view, filter, and search for users by their Username, Display Name, or Email.

Automated Password Change on Import

An option has been added to the Import Secret function to mark each secret's password to immediately change after import. With this option enabled, a user who has had access to view the list of secrets will no longer know the password of the secrets once they have completed the import. The option is available for CSV and XML and can be flagged via the UI and API.

Enhancements

11.6.000002 GA

- Improvement: We made Web HTML elements IDs unique to prevent conflicts and keep buttons accessible.
- Improvement: Enhanced the user experience when navigating Secret Server with no permissions
- Improvement: Added filtering of the Description field of the Discovery Network View when entering search text.

- Improvement: Increased the specificity of an exception when accessing the REST API without permission—it now returns AccessDeniedException instead of API_AccessDenied.

11.6.000000 EA

- Improvement: User tooltips in both Secret Server and Delinea Platform now highlight the Platform Integration Types.
- Improvement: (Disaster Recovery/Resilient Secrets):Data replication will now create personal folders for replicated users in cases where the replica blocks or does not allow personal folders to be replicated. This is only if personal folders are enabled on the replica.
- Improvement: User tooltips in both Secret Server and Delinea Platform now highlight the Platform Integration Types.
- Improvement: Fixed issues with user and group syncing between Secret Server Cloud and the Delinea Platform.
- Improvement: Added a "Managed" field to the Discovery Network view to show when a discovery item is managed.
- Improvement: The Password Requirement Audit has been converted to the new UI.
- Improvement: The Secret Dependency Changers editor has been converted to the new UI.
- Improvement: Dependency Templates are now available in the new UI.
- Improvement: Session playback player UI has been updated.
- Improvement: The Launcher Audits page has been migrated to the new UI.
- Improvement: Discovery Service Accounts Detail Page now shows services that run as the directory account as well as the computers on which that service runs
- Improvement: Added a Quick Access link to see all secrets you currently have checked out.
- Improvement: Updated Createuser.aspx to redirect to the new user creation page.
- Improvement: Updated the group role assignment UI.
- Improvement: Group membership assignment UI updated.
- Improvement: Group role assignment UI updated.
- Improvement: Updated process for populating a forthcoming computer-centric view.
- Improvement: Session recording search now uses updated filter pattern.
- Improvement: The built-in "Everyone" group was renamed "All Vault Users."
- Improvement: Enhanced new Discovery Area to include some additional fields and added logic for the error chip being displayed
- Improvement: Added a Copy button for Data Source URL on Disaster Recovery - Outgoing Setup Steps modal.
- Improvement: New Vault User Details in the Platform overview for Users tab. It requires a Vault to be successfully connected and configured for the details to appear, otherwise the section does not appear.

- Improvement: Added banners to various Roles/Permissions pages in Secret Server Cloud and Platform with links to help navigate between the two.
- Improvement: Secret Share tab UI has been updated to match the permission setting experience for setting folder permissions. Domain name is now displayed for users on the secret share tab.
- Improvement: Fixed an issue where the folder permissions tab would load slowly with large numbers of users.
- Improvement: Updated group membership management pages to use new design patterns.
- Improvement: The display name of the secret Vault is now set via the Platform. The Vault subcategories for Reporting, Inbox, and administration have been updated to reflect Secret Server.
- Improvement: Analysis tab of Discovery no longer includes disabled Discovery Sources in managed/unmanaged counts.
- Improvement: Administration Configuration Launcher Settings now displays the Enable Protocol Handler Auto-Update setting in cloud.
- Improvement: View Log was hidden for Directory Accounts since there's no computer associated to show the log of.
- Improvement: Added Application from tbAuditSecret to session search results model and session model.
- Improvement: When discovery is running the network view performance would timeout depending on SQL locks. This should no longer happen.
- Improvement: Discovery scanners added an option to "Add child scanner" which filters available scanners to show only applicable child scanners.
- Improvement: Disaster Recovery Add-On Licensing handling added.
- Improvement: Secret template fields table has been updated and has an improved drag and drop experience.
- Improvement: Secret panel is more mobile friendly.
- Improvement: Syslog/CEF logging enhanced to capture more detailed metadata for secrets that contain fields/data that map to the following SIEM fields: Account Name, Account Domain, Target Server, Request ID (that is from Ticketing System). Additionally, failed attempts to access secrets due to Ticket Validation errors are now also logged to Secret Audits.
- Improvement: New inbox notification bell with panel, allows for viewing and approving inbox items without having to navigate through the site.
- Improvement: The Security Audit Log page has been converted to the latest UI.
- Improvement: A donut chart showing different Operating Systems in discovery has been added to the Analysis tab of discovery.
- Improvement: Live viewing has been added to the new session monitoring.
- Improvement: The new UI Discovery Rules page now shows the correct Secret Template name.
- Improvement: Secret policy now links to the policy on the secret general tab.
- Improvement: A loading indicator now shows when opening the discovery add scanner dialog.

- Improvement: The main top left logo will link to the users preferred login home if it is the dashboard or all secrets.
- Improvement: The COM+ scanner will be able to be added, but there will be a note in the preview panel letting the user know that the scanner will not work for a site that is set to UseWebsite.
- Improvement: A preview chip has been added to Multifactor Authentication on Secrets and its supporting configuration pages.
- Improvement: A new field "Full Name" has been added to the discovery network view to give a more detailed version of the item's name.
- Improvement: Default columns have been added per Item Type in the discovery network view.
- Improvement: Dependency Tokens are now available on the dependency edit screen.
- Improvement: Enhanced loading times of Secret Server elements in Delinea Platform.
- Improvement: REST API documentation has links to individual services that load quickly.
- Improvement: Added filter on recorded-sessions endpoint to filter out applications, particularly 'RemoteAccessService' when in platform
- Improvement: Implemented a message shim in the Vault Broker to inform Secret Server that a user's platform permissions have changed.
- Improvement: Updated the Vault Settings and Vault User Detail Tabs with some UI changes.
- Improvement: Converted the creation of a Password Changer when Create Password Changer is selected from the Password Changers list in Remote Password Changing.
- Improvement: Added a filter of secretIDs to the Secret Search endpoint to that Secrets can be filtered by SecretID.
- Improvement: Terminate, limit to 5 minutes, and message only have been added to live viewing in the new session monitoring
- Improvement: The heading for Vault within Platform User Management details has been updated to read its value from within Platform.
- Improvement: The text for page title, breadcrumbs, and navigation for Secret Server Reporting have been updated in Platform to match.
- Improvement: Added Search Groups column to Discovery Network View.
- Improvement: Added more instructions regarding Disaster Recovery's data storage path configuration setting.
- Improvement: Added configuration setting to determine which secret permission is required to change Remote Password Changing settings on a Secret. Owner or Edit.
- Improvement: The breadcrumbs within the RPC administration pages have been standardized. The links within Platform Vault Configuration Overview no longer cause the page to reload.
- Improvement: Added a WMI Service Timeout setting to the cloud advanced configuration page to help with dependency changes that take more time than the allotted 60 seconds.
- Improvement: The PowerShell script timeout no longer defaults to 90 seconds. Instead, it now uses the value from the Event Pipelines Maximum Script Run Time (Minutes) setting in advanced configuration.

- Improvement: Improved performance of Secret Search for customers with large numbers of Secrets.
- Improvement: Updated data type to support frequent users of session recording that was crashing the encoding process.
- Improvement: Secrets with text field based URL lists are now searchable.
- Improvement: Platform users can login to Terminal using SSH Key Integration.
- Improvement: Added validation messages to password requirement rules for when password requirements are too complex to reliably generate a password.
- Improvement: When Platform integration is active the integration page will now have a button to reset mappings from Delinea Platform.
- Improvement: AD Privilege Password changer now has Remote Password Change timeout minutes Advanced Setting.
- Improvement: Better handling of unexpected heartbeat behavior to mitigate reported Distributed Engine stalling.
- Improvement: Connect As Credentials on Secret works better with SSH Keys for su user switching.
- Improvement: Updated links on the Security Hardening Report to new UI pages
- Improvement: When creating a new send to syslog task you no longer get a default schedule. Most of the templates didn't create a schedule, now they're all consistent.
- Improvement: Session monitoring search now supports searching by a single secret.
- Improvement: When a Secret is assigned to a site the user does not have access to due to Teams restriction, they will see the word "Restricted" instead of "Site Name (Inactive)".
- Improvement: Mitigated issue in large bulk secret actions.
- Improvements: The "Synchronization Running" message for DR will now only appear if there is a recorded start time for DR in the past and a finish time that is in the future.
- Improvement: Added Secret Field validation on the Template level to ensure users cannot create a "Secret Name" field on a template.
- Improvement: Default values for Secret Fields such as port will now be replicated for Disaster Recovery.
- Improvement: A user with only direct access to a report and the "browse reports" role permission can add that report to the dashboard.
- Improvement: The breadcrumbs within the RPC administration pages have been standardized. The links within Platform Vault Configuration Overview no longer cause the page to reload.
- Improvement: Report column preferences will be saved and applied when viewing a report.
- Improvement: The Secrets grid now updates displayed data and selected columns simultaneously.
- Improvement: Improved error logging and efficiency for calls coming from Delinea Platform.
- Improvement: Quick access filters now both apply when updated.
- Improvement: Knowledge base links within Platform Vault now link to their intended location.

- Improvement: Corrected edge case that could result in a session view audit being placed on the incorrect Secret.
- Improvement: The Parent Scan Template will be filtered to the type and will default to the first item in the list on create. The proper fields will be shown based on the type.
- Improvement: If a secret is inactivated after initially viewing the secret, a user that cannot view inactive secrets will no longer get an error from secret heartbeat.
- Improvement: Clicking cancel when editing folder permissions will clear any active filters.
- Improvement: Editing folder permissions now has a split button that allows for directly entering edit or add group/user mode.
- Improvement: The Secrets Quick Access link when collapsed, now targets the correct destination.
- Improvement: The Platform Opt In modal styling has been adjusted to no longer display with scroll bars.
- Improvement: Secret Share and Folder Permissions: Show disabled edit button until filters are loaded since split button does not yet support disabled.
- Improvement: API calls to /v[1/2]/secrets/{id} now update the Recents secrets data source.
- Improvement: When viewing Event Pipeline Activity details, selecting an Activity Detail record from the grid now displays the selected Activity's details.
- Improvement: Added query parameter for PipelineId to pass back when viewing specific pipeline activity
- Improvement: Discovery Scanner will not allow deletion until Secret selection is changed.
- Improvement: Remote Password Changing: Check for DNS Mismatch now visible and functional in Cloud.
- Improvement: EventTime token is available in pipeline scripts. $EventTime - event date and time of the event ("yyyy'-'MM'-'dd'T'HH':'mm':'ss").
- Improvement: The preview chips for Multifactor on Secrets have been removed.
- Improvement: Creating a User SSH Key in Platform downloads the private key with a proper filename.
- Improvement: Cipher Suite Configuration now allows configuration of allowed Host Key Algorithms.

Bug Fixes

11.6.000002 GA

- Placed part of the "secret save" process into a transaction so that changes would be rolled back if a timeout occurred.
- Fixed a bug that prevented expanding a secret created using a custom template on All Secrets View.
- Added a bulk operation to set password requirements on multiple secrets.
- Added a "password displayed" audit when viewing a secret transition history.
- Allowed AutoChangeSchedule to be usable when CheckoutChangePassword is enabled.
- Fixed the Inbox link in the left navigation panel.

- Extended the "secret hook" timeout from 30 seconds to 2 minutes. Hooks now use the "Event Pipelines Maximum Script Run Time (Minutes)" advanced setting to extend beyond 2 minutes.
- Fixed a localization issue when a discovery item has a scan item template that is not out of the box.
- Fixed issue with secret permissions displaying incorrectly on the Secret Share page.
- 11.6.000000 EA
- On-premise only: Fixed logic issue for on-premise instance that was using the wrong minimum heartbeat interval. Minimum heartbeat interval set back to 5 minutes.

  Fixed an issue where Viewing Session Connector Custom Launchers without access to the RDS Credentials secret would show an error.
- Fixed an issue where unplayable session recording videos would display an infinite load instead of the appropriate error.
- Fixed an issue where a secret template could be saved without RPC mappings configured.
- Fixed an issue where Pause times for ODBC Remote Password Changers were not adhered to. If you feel your RPC's are running slowly, check the pause times and remove them if they are not needed for the RPC action.
- Fixed an issue where Web Password Filler didn't work in certain instances due to an ambiguity in interpreting the Secret Server URL.
- Fixed an issue where setting custom expiration dates in all time zones did not work correctly.
- Fixed an issue where all event subscriptions did not fire for secrets in subfolders of the target folder.
- Fixed an issue where the secret name would incorrectly display on the New Discovery Import Rules page.
- Fixed an issue with negative numbers exporting incorrectly when exporting to a CSV file.
- Fixed an issue where a large number of SSH terminal connection history records causing timeouts.
- Fixed an issue with hidden days until deletion field when enabling deletion in the retention schedule. Added localization to error when trying to submit days less than or equal to the archive retention value.
- Fixed an issue with passwords being uneditable if RPC is set to use a Privileged Secret to which the user has no access to. Restored explanatory banner.
- Fixed an issue where secrets aren't synced with DevOps in cloud with when triggered by pipelines.
- Fixed issue in discovery where computer scans were sometimes throwing string truncation exceptions.
- Fixed an issue where TOTP Secret Settings edit button was available to users who could not edit the settings.
- Fix an issue with editing Session Connector Custom Launcher Port.
- Fixed a UI issue with the launcher popup window showing an option the user didn't have permission for.
- Fixed an issue where configuring a new session connector launcher might not show all available launcher types.
- Fixed an issue where configuring "Use Additional Prompt" on launchers might prevent save.
- Fixed an issue with the TemplateCreateSecret role link.
- Fixed an issue with View Launcher Password.

- Fixed an issue where users with only 'View' access on a Secret would be unable to view the Password if there was a custom launcher with arguments configured for that Secret Template.
- Fixed an issue with DSV sync for secret with file type fields and no file set.
- Fixed an issue with localization on folder Metadata page.
- Fixed an issue with sorting for Checkout User Id and Checkout User.
- Fixed an issue with ODBC password changing that broke postgres and mySQL changing.
- Fixed a logging issue with Dependency changes ran through Distributed Engine being skipped due to conditions.
- Fixed an issue where the generate SSH key returns a 500 exception.
- Fixed an issue where the SSHCipherSuiteModel GetAsync returned a 500 exception.
- Fixed an issue where the CreatePublicSSHKey returned a 500 Exception.
- Fixed an issue where Discovery Scanners could not be removed until the associated secrets had been edited.

Secret Server 11.6.000002 GA Release Notes

For convenience, this document also contains the release notes from the 11.6.000000 EA release.

Release Date and Notes

On-premises: September 26, 2023

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.16.0

Protocol Handler: 6.0.3.26

Note: Step Upgrade Required (11.5.2). Versions prior to 11.5.2 will need to first upgrade to 11.5.2. The automatic downloads in the product will get the right versions for the step upgrade and then allow the 11.6 upgrade. But if offline and using the file upload method, versions prior to 11.5.2 will get an error message saying, "Integrity Check failed - Security Catalog is signed by thumbprint that is not specifically trusted." The remedy is to first upgrade to 11.5.2 and then do the upgrade to 11.6.

Feature Enhancements

View Own Session Recordings Permission

A new permission has been added "View Own Session Recordings". With this permission, a user can be restricted to only viewing the recordings that they initiated. If the user with this permission clicks on a recording initiated and owned by another user, they will get an Access Denied window. In addition, the "View Session Recordings" permission has been renamed to "View All Session Recordings".

Session Monitoring Playback Page UI Conversion

The Session Recording playback page has been updated to the new UI, including a new video player with additional controls. The SSH keystroke-only playback page and the video playback page have been merged, and the available elements will be shown. The legacy player is still available as a link from the new video playback page.

Note: The activity graph and download button have not yet been implemented for the new page, although they are available on the legacy page if needed.

Manual Password Change for Checked Out Secrets

Secrets with Change Password on Checkin configured now have the "Change Password Now" functionality available. This will enable the standard functionality of a password change, and the secret will also complete the automatic password change on checking in. This is to allow maintenance and testing of secrets protected in this manner, and a pending password change must be completed before the check-in process is allowed to begin in order to maintain a secure order of operations.

Updated User Selection Interface

Various locations around the product provide a user selection interface to provide the ability to select a user as the target of a particular configuration, such as the permissions, groups, and roles pages. These have been updated throughout the product to provide more data about the users in the list. You may now view, filter, and search for users by their Username, Display Name, or Email.

Automated Password Change on Import

An option has been added to the Import Secret function to mark each secret's password to immediately change after import. With this option enabled, a user who has had access to view the list of secrets will no longer know the password of the secrets once they have completed the import. The option is available for CSV and XML and can be flagged via the UI and API.

Enhancements

11.6.000002 GA

- Improvement: We made Web HTML elements IDs unique to prevent conflicts and keep buttons accessible.
- Improvement: Enhanced the user experience when navigating Secret Server with no permissions
- Improvement: Added filtering of the Description field of the Discovery Network View when entering search text.
- Improvement: Increased the specificity of an exception when accessing the REST API without permission—it now returns AccessDeniedException instead of API_AccessDenied.

11.6.000000 EA

- Improvement: User tooltips in both Secret Server and Delinea Platform now highlight the Platform Integration Types.
- Improvement: (Disaster Recovery/Resilient Secrets):Data replication will now create personal folders for replicated users in cases where the replica blocks or does not allow personal folders to be replicated. This is only if personal folders are enabled on the replica.
- Improvement: User tooltips in both Secret Server and Delinea Platform now highlight the Platform Integration Types.
- Improvement: Fixed issues with user and group syncing between Secret Server Cloud and the Delinea Platform.
- Improvement: Added a "Managed" field to the Discovery Network view to show when a discovery item is managed.
- Improvement: The Password Requirement Audit has been converted to the new UI.
- Improvement: The Secret Dependency Changers editor has been converted to the new UI.
- Improvement: Dependency Templates are now available in the new UI.
- Improvement: Session playback player UI has been updated.
- Improvement: The Launcher Audits page has been migrated to the new UI.
- Improvement: Discovery Service Accounts Detail Page now shows services that run as the directory account as well as the computers on which that service runs
- Improvement: Added a Quick Access link to see all secrets you currently have checked out.
- Improvement: Updated Createuser.aspx to redirect to the new user creation page.
- Improvement: Updated the group role assignment UI.
- Improvement: Group membership assignment UI updated.
- Improvement: Group role assignment UI updated.
- Improvement: Updated process for populating a forthcoming computer-centric view.
- Improvement: Session recording search now uses updated filter pattern.

- Improvement: The built-in "Everyone" group was renamed "All Vault Users."
- Improvement: Enhanced new Discovery Area to include some additional fields and added logic for the error chip being displayed
- Improvement: Added a Copy button for Data Source URL on Disaster Recovery - Outgoing Setup Steps modal.
- Improvement: New Vault User Details in the Platform overview for Users tab. It requires a Vault to be successfully connected and configured for the details to appear, otherwise the section does not appear.
- Improvement: Added banners to various Roles/Permissions pages in Secret Server Cloud and Platform with links to help navigate between the two.
- Improvement: Secret Share tab UI has been updated to match the permission setting experience for setting folder permissions. Domain name is now displayed for users on the secret share tab.
- Improvement: Fixed an issue where the folder permissions tab would load slowly with large numbers of users.
- Improvement: Updated group membership management pages to use new design patterns.
- Improvement: The display name of the secret Vault is now set via the Platform. The Vault subcategories for Reporting, Inbox, and administration have been updated to reflect Secret Server.
- Improvement: Analysis tab of Discovery no longer includes disabled Discovery Sources in managed/unmanaged counts.
- Improvement: Administration Configuration Launcher Settings now displays the Enable Protocol Handler Auto-Update setting in cloud.
- Improvement: View Log was hidden for Directory Accounts since there's no computer associated to show the log of.
- Improvement: Added Application from tbAuditSecret to session search results model and session model.
- Improvement: When discovery is running the network view performance would timeout depending on SQL locks. This should no longer happen.
- Improvement: Discovery scanners added an option to "Add child scanner" which filters available scanners to show only applicable child scanners.
- Improvement: Disaster Recovery Add-On Licensing handling added.
- Improvement: Secret template fields table has been updated and has an improved drag and drop experience.
- Improvement: Secret panel is more mobile friendly.
- Improvement: Syslog/CEF logging enhanced to capture more detailed metadata for secrets that contain fields/data that map to the following SIEM fields: Account Name, Account Domain, Target Server, Request ID (that is from Ticketing System). Additionally, failed attempts to access secrets due to Ticket Validation errors are now also logged to Secret Audits.
- Improvement: New inbox notification bell with panel, allows for viewing and approving inbox items without having to navigate through the site.
- Improvement: The Security Audit Log page has been converted to the latest UI.
- Improvement: A donut chart showing different Operating Systems in discovery has been added to the Analysis tab of discovery.

- Improvement: Live viewing has been added to the new session monitoring.
- Improvement: The new UI Discovery Rules page now shows the correct Secret Template name.
- Improvement: Secret policy now links to the policy on the secret general tab.
- Improvement: A loading indicator now shows when opening the discovery add scanner dialog.
- Improvement: The main top left logo will link to the users preferred login home if it is the dashboard or all secrets.
- Improvement: The COM+ scanner will be able to be added, but there will be a note in the preview panel letting the user know that the scanner will not work for a site that is set to UseWebsite.
- Improvement: A preview chip has been added to Multifactor Authentication on Secrets and its supporting configuration pages.
- Improvement: A new field "Full Name" has been added to the discovery network view to give a more detailed version of the item's name.
- Improvement: Default columns have been added per Item Type in the discovery network view.
- Improvement: Dependency Tokens are now available on the dependency edit screen.
- Improvement: Enhanced loading times of Secret Server elements in Delinea Platform.
- Improvement: REST API documentation has links to individual services that load quickly.
- Improvement: Added filter on recorded-sessions endpoint to filter out applications, particularly 'RemoteAccessService' when in platform
- Improvement: Implemented a message shim in the Vault Broker to inform Secret Server that a user's platform permissions have changed.
- Improvement: Updated the Vault Settings and Vault User Detail Tabs with some UI changes.
- Improvement: Converted the creation of a Password Changer when Create Password Changer is selected from the Password Changers list in Remote Password Changing.
- Improvement: Added a filter of secretIDs to the Secret Search endpoint to that Secrets can be filtered by SecretID.
- Improvement: Terminate, limit to 5 minutes, and message only have been added to live viewing in the new session monitoring
- Improvement: The heading for Vault within Platform User Management details has been updated to read its value from within Platform.
- Improvement: The text for page title, breadcrumbs, and navigation for Secret Server Reporting have been updated in Platform to match.
- Improvement: Added Search Groups column to Discovery Network View.
- Improvement: Added more instructions regarding Disaster Recovery's data storage path configuration setting.
- Improvement: Added configuration setting to determine which secret permission is required to change Remote Password Changing settings on a Secret. Owner or Edit.

- Improvement: The breadcrumbs within the RPC administration pages have been standardized. The links within Platform Vault Configuration Overview no longer cause the page to reload.
- Improvement: Added a WMI Service Timeout setting to the cloud advanced configuration page to help with dependency changes that take more time than the allotted 60 seconds.
- Improvement: The PowerShell script timeout no longer defaults to 90 seconds. Instead, it now uses the value from the Event Pipelines Maximum Script Run Time (Minutes) setting in advanced configuration.
- Improvement: Improved performance of Secret Search for customers with large numbers of Secrets.
- Improvement: Updated data type to support frequent users of session recording that was crashing the encoding process.
- Improvement: Secrets with text field based URL lists are now searchable.
- Improvement: Platform users can login to Terminal using SSH Key Integration.
- Improvement: Added validation messages to password requirement rules for when password requirements are too complex to reliably generate a password.
- Improvement: When Platform integration is active the integration page will now have a button to reset mappings from Delinea Platform.
- Improvement: AD Privilege Password changer now has Remote Password Change timeout minutes Advanced Setting.
- Improvement: Better handling of unexpected heartbeat behavior to mitigate reported Distributed Engine stalling.
- Improvement: Connect As Credentials on Secret works better with SSH Keys for su user switching.
- Improvement: Updated links on the Security Hardening Report to new UI pages
- Improvement: When creating a new send to syslog task you no longer get a default schedule. Most of the templates didn't create a schedule, now they're all consistent.
- Improvement: Session monitoring search now supports searching by a single secret.
- Improvement: When a Secret is assigned to a site the user does not have access to due to Teams restriction, they will see the word "Restricted" instead of "Site Name (Inactive)".
- Improvement: Mitigated issue in large bulk secret actions.
- Improvements: The "Synchronization Running" message for DR will now only appear if there is a recorded start time for DR in the past and a finish time that is in the future.
- Improvement: Added Secret Field validation on the Template level to ensure users cannot create a "Secret Name" field on a template.
- Improvement: Default values for Secret Fields such as port will now be replicated for Disaster Recovery.
- Improvement: A user with only direct access to a report and the "browse reports" role permission can add that report to the dashboard.
- Improvement: The breadcrumbs within the RPC administration pages have been standardized. The links within Platform Vault Configuration Overview no longer cause the page to reload.
- Improvement: Report column preferences will be saved and applied when viewing a report.

- Improvement: The Secrets grid now updates displayed data and selected columns simultaneously.
- Improvement: Improved error logging and efficiency for calls coming from Delinea Platform.
- Improvement: Quick access filters now both apply when updated.
- Improvement: Knowledge base links within Platform Vault now link to their intended location.
- Improvement: Corrected edge case that could result in a session view audit being placed on the incorrect Secret.
- Improvement: The Parent Scan Template will be filtered to the type and will default to the first item in the list on create. The proper fields will be shown based on the type.
- Improvement: If a secret is inactivated after initially viewing the secret, a user that cannot view inactive secrets will no longer get an error from secret heartbeat.
- Improvement: Clicking cancel when editing folder permissions will clear any active filters.
- Improvement: Editing folder permissions now has a split button that allows for directly entering edit or add group/user mode.
- Improvement: The Secrets Quick Access link when collapsed, now targets the correct destination.
- Improvement: The Platform Opt In modal styling has been adjusted to no longer display with scroll bars.
- Improvement: Secret Share and Folder Permissions: Show disabled edit button until filters are loaded since split button does not yet support disabled.
- Improvement: API calls to /v[1/2]/secrets/{id} now update the Recents secrets data source.
- Improvement: When viewing Event Pipeline Activity details, selecting an Activity Detail record from the grid now displays the selected Activity's details.
- Improvement: Added query parameter for PipelineId to pass back when viewing specific pipeline activity
- Improvement: Minimum Heartbeat interval reduced from 15 to 5 minutes.
- Improvement: Discovery Scanner will not allow deletion until Secret selection is changed.
- Improvement: Remote Password Changing: Check for DNS Mismatch now visible and functional in Cloud.
- Improvement: EventTime token is available in pipeline scripts. $EventTime - event date and time of the event ("yyyy'-'MM'-'dd'T'HH':'mm':'ss").
- Improvement: The preview chips for Multifactor on Secrets have been removed.
- Improvement: Creating a User SSH Key in Platform downloads the private key with a proper filename.
- Improvement: Cipher Suite Configuration now allows configuration of allowed Host Key Algorithms.

Bug Fixes

11.000002 GA

- Placed part of the "secret save" process into a transaction so that changes would be rolled back if a timeout occurred.
- Fixed a bug that prevented expanding a secret created using a custom template on All Secrets View.
- Added a bulk operation to set password requirements on multiple secrets.

- Added a "password displayed" audit when viewing a secret transition history.
- Allowed AutoChangeSchedule to be usable when CheckoutChangePassword is enabled.
- Fixed the Inbox link in the left navigation panel.
- Extended the "secret hook" timeout from 30 seconds to 2 minutes. Hooks now use the "Event Pipelines Maximum Script Run Time (Minutes)" advanced setting to extend beyond 2 minutes.
- Fixed a localization issue when a discovery item has a scan item template that is not out of the box.
- Fixed issue with secret permissions displaying incorrectly on the Secret Share page.

- 11.000000 EA

- Fixed an issue where Viewing Session Connector Custom Launchers without access to the RDS Credentials secret would show an error.
- Fixed an issue where unplayable session recording videos would display an infinite load instead of the appropriate error.
- Fixed an issue where a secret template could be saved without RPC mappings configured.
- Fixed an issue where Pause times for ODBC Remote Password Changers were not adhered to. If you feel your RPC's are running slowly, check the pause times and remove them if they are not needed for the RPC action.
- Fixed an issue where Web Password Filler didn't work in certain instances due to an ambiguity in interpreting the Secret Server URL.
- Fixed an issue where setting custom expiration dates in all time zones did not work correctly.
- Fixed an issue where all event subscriptions did not fire for secrets in subfolders of the target folder.
- Fixed an issue where the secret name would incorrectly display on the New Discovery Import Rules page.
- Fixed an issue with negative numbers exporting incorrectly when exporting to a CSV file.
- Fixed an issue where a large number of SSH terminal connection history records causing timeouts.
- Fixed an issue with hidden days until deletion field when enabling deletion in the retention schedule. Added localization to error when trying to submit days less than or equal to the archive retention value.
- Fixed an issue with passwords being uneditable if RPC is set to use a Privileged Secret to which the user has no access to. Restored explanatory banner.
- Fixed an issue where secrets aren't synced with DevOps in cloud with when triggered by pipelines.
- Fixed issue in discovery where computer scans were sometimes throwing string truncation exceptions.
- Fixed an issue where TOTP Secret Settings edit button was available to users who could not edit the settings.
- Fix an issue with editing Session Connector Custom Launcher Port.
- Fixed a UI issue with the launcher popup window showing an option the user didn't have permission for.
- Fixed an issue where configuring a new session connector launcher might not show all available launcher types.
- Fixed an issue where configuring "Use Additional Prompt" on launchers might prevent save.

- Fixed an issue with the TemplateCreateSecret role link.
- Fixed an issue with View Launcher Password.
- Fixed an issue where users with only 'View' access on a Secret would be unable to view the Password if there was a custom launcher with arguments configured for that Secret Template.
- Fixed an issue with DSV sync for secret with file type fields and no file set.
- Fixed an issue with localization on folder Metadata page.
- Fixed an issue with sorting for Checkout User Id and Checkout User.
- Fixed an issue with ODBC password changing that broke postgres and mySQL changing.
- Fixed a logging issue with Dependency changes ran through Distributed Engine being skipped due to conditions.
- Fixed an issue where the generate SSH key returns a 500 exception.
- Fixed an issue where the SSHCipherSuiteModel GetAsync returned a 500 exception.
- Fixed an issue where the CreatePublicSSHKey returned a 500 Exception.
- Fixed an issue where Discovery Scanners could not be removed until the associated secrets had been edited.

Secret Server 11.6.000000 EA Release Notes

Release Date and Notes

On-premises: September 12, 2023

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.16.0

Protocol Handler: 6.0.3.26

Step Upgrade Required (11.5.2). Versions prior to 11.5.2 will need to first upgrade to 11.5.2. The automatic downloads in the product will get the right versions for the step upgrade and then allow the 11.6 upgrade. But if offline and using the file upload method, versions prior to 11.5.2 will get an error message saying, "Integrity Check failed - Security Catalog is signed by thumbprint that is not specifically trusted." The remedy is to first upgrade to 11.5.2 and then do the upgrade to 11.6. For instructions on upgrading in general, go to "Upgrading" on page 124.

Feature Enhancements

View Own Session Recordings Permission

A new permission has been added "View Own Session Recordings". With this permission, a user can be restricted to only viewing the recordings that they initiated. If the user with this permission clicks on a recording initiated and owned by another user, they will get an Access Denied window. In addition, the "View Session Recordings" permission has been renamed to "View All Session Recordings".

Session Monitoring Playback Page UI Conversion

The Session Recording playback page has been updated to the new UI, including a new video player with additional controls. The SSH keystroke-only playback page and the video playback page have been merged, and the available elements will be shown. The legacy player is still available as a link from the new video playback page.

Note: The activity graph and download button have not yet been implemented for the new page, although they are available on the legacy page if needed.

Manual Password Change for Checked Out Secrets

Secrets with Change Password on Checkin configured now have the "Change Password Now" functionality available. This will enable the standard functionality of a password change, and the secret will also complete the automatic password change on checking in. This is to allow maintenance and testing of secrets protected in this manner, and a pending password change must be completed before the check-in process is allowed to begin in order to maintain a secure order of operations.

Updated User Selection Interface

Various locations around the product provide a user selection interface to provide the ability to select a user as the target of a particular configuration, such as the permissions, groups, and roles pages. These have been updated throughout the product to provide more data about the users in the list. You may now view, filter, and search for users by their Username, Display Name, or Email.

Automated Password Change on Import

An option has been added to the Import Secret function to mark each secret's password to immediately change after import. With this option enabled, a user who has had access to view the list of secrets will no longer know the password of the secrets once they have completed the import. The option is available for CSV and XML and can be flagged via the UI and API.

Enhancements

- Improvement: User tooltips in both Secret Server and Delinea Platform now highlight the Platform Integration Types.
- Improvement: (Disaster Recovery/Resilient Secrets):Data replication will now create personal folders for replicated users in cases where the replica blocks or does not allow personal folders to be replicated. This is only if personal folders are enabled on the replica.
- Improvement: User tooltips in both Secret Server and Delinea Platform now highlight the Platform Integration Types.
- Improvement: Fixed issues with user and group syncing between Secret Server Cloud and the Delinea Platform.
- Improvement: Added a "Managed" field to the Discovery Network view to show when a discovery item is managed.
- Improvement: The Password Requirement Audit has been converted to the new UI.
- Improvement: The Secret Dependency Changers editor has been converted to the new UI.
- Improvement: Dependency Templates are now available in the new UI.
- Improvement: Session playback player UI has been updated.
- Improvement: The Launcher Audits page has been migrated to the new UI.
- Improvement: Discovery Service Accounts Detail Page now shows services that run as the directory account as well as the computers on which that service runs
- Improvement: Added a Quick Access link to see all secrets you currently have checked out.
- Improvement: Updated Createuser.aspx to redirect to the new user creation page.
- Improvement: Updated the group role assignment UI.
- Improvement: Group membership assignment UI updated.
- Improvement: Group role assignment UI updated.
- Improvement: Updated process for populating a forthcoming computer-centric view.
- Improvement: Session recording search now uses updated filter pattern.
- Improvement: The built-in "Everyone" group was renamed "All Vault Users."
- Improvement: Enhanced new Discovery Area to include some additional fields and added logic for the error chip being displayed
- Improvement: Added a Copy button for Data Source URL on Disaster Recovery - Outgoing Setup Steps modal.
- Improvement: New Vault User Details in the Platform overview for Users tab. It requires a Vault to be successfully connected and configured for the details to appear, otherwise the section does not appear.
- Improvement: Added banners to various Roles/Permissions pages in Secret Server Cloud and Platform with links to help navigate between the two.
- Improvement: Secret Share tab UI has been updated to match the permission setting experience for setting folder permissions. Domain name is now displayed for users on the secret share tab.

- Improvement: Fixed an issue where the folder permissions tab would load slowly with large numbers of users.
- Improvement: Updated group membership management pages to use new design patterns.
- Improvement: The display name of the secret Vault is now set via the Platform. The Vault subcategories for Reporting, Inbox, and administration have been updated to reflect Secret Server.
- Improvement: Analysis tab of Discovery no longer includes disabled Discovery Sources in managed/unmanaged counts.
- Improvement: Administration Configuration Launcher Settings now displays the Enable Protocol Handler Auto-Update setting in cloud.
- Improvement: View Log was hidden for Directory Accounts since there's no computer associated to show the log of.
- Improvement: Added Application from tbAuditSecret to session search results model and session model.
- Improvement: When discovery is running the network view performance would timeout depending on SQL locks. This should no longer happen.
- Improvement: Discovery scanners added an option to "Add child scanner" which filters available scanners to show only applicable child scanners.
- Improvement: Disaster Recovery Add-On Licensing handling added.
- Improvement: Secret template fields table has been updated and has an improved drag and drop experience.
- Improvement: Secret panel is more mobile friendly.
- Improvement: Syslog/CEF logging enhanced to capture more detailed metadata for secrets that contain fields/data that map to the following SIEM fields: Account Name, Account Domain, Target Server, Request ID (that is from Ticketing System). Additionally, failed attempts to access secrets due to Ticket Validation errors are now also logged to Secret Audits.
- Improvement: New inbox notification bell with panel, allows for viewing and approving inbox items without having to navigate through the site.
- Improvement: The Security Audit Log page has been converted to the latest UI.
- Improvement: A donut chart showing different Operating Systems in discovery has been added to the Analysis tab of discovery.
- Improvement: Live viewing has been added to the new session monitoring.
- Improvement: The new UI Discovery Rules page now shows the correct Secret Template name.
- Improvement: Secret policy now links to the policy on the secret general tab.
- Improvement: A loading indicator now shows when opening the discovery add scanner dialog.
- Improvement: The main top left logo will link to the users preferred login home if it is the dashboard or all secrets.
- Improvement: The COM+ scanner will be able to be added, but there will be a note in the preview panel letting the user know that the scanner will not work for a site that is set to UseWebsite.

- Improvement: A preview chip has been added to Multifactor Authentication on Secrets and its supporting configuration pages.
- Improvement: A new field "Full Name" has been added to the discovery network view to give a more detailed version of the item's name.
- Improvement: Default columns have been added per Item Type in the discovery network view.
- Improvement: Dependency Tokens are now available on the dependency edit screen.
- Improvement: Enhanced loading times of Secret Server elements in Delinea Platform.
- Improvement: REST API documentation has links to individual services that load quickly.
- Improvement: Added filter on recorded-sessions endpoint to filter out applications, particularly 'RemoteAccessService' when in platform
- Improvement: Implemented a message shim in the Vault Broker to inform Secret Server that a user's platform permissions have changed.
- Improvement: Updated the Vault Settings and Vault User Detail Tabs with some UI changes.
- Improvement: Converted the creation of a Password Changer when Create Password Changer is selected from the Password Changers list in Remote Password Changing.
- Improvement: Added a filter of secretIDs to the Secret Search endpoint to that Secrets can be filtered by SecretID.
- Improvement: Terminate, limit to 5 minutes, and message only have been added to live viewing in the new session monitoring
- Improvement: The heading for Vault within Platform User Management details has been updated to read its value from within Platform.
- Improvement: The text for page title, breadcrumbs, and navigation for Secret Server Reporting have been updated in Platform to match.
- Improvement: Added Search Groups column to Discovery Network View.
- Improvement: Added more instructions regarding Disaster Recovery's data storage path configuration setting.
- Improvement: Added configuration setting to determine which secret permission is required to change Remote Password Changing settings on a Secret. Owner or Edit.
- Improvement: The breadcrumbs within the RPC administration pages have been standardized. The links within Platform Vault Configuration Overview no longer cause the page to reload.
- Improvement: Added a WMI Service Timeout setting to the cloud advanced configuration page to help with dependency changes that take more time than the allotted 60 seconds.
- Improvement: The PowerShell script timeout no longer defaults to 90 seconds. Instead, it now uses the value from the Event Pipelines Maximum Script Run Time (Minutes) setting in advanced configuration.
- Improvement: Improved performance of Secret Search for customers with large numbers of Secrets.
- Improvement: Updated data type to support frequent users of session recording that was crashing the encoding process.
- Improvement: Secrets with text field based URL lists are now searchable.

- Improvement: Platform users can login to Terminal using SSH Key Integration.
- Improvement: Added validation messages to password requirement rules for when password requirements are too complex to reliably generate a password.
- Improvement: When Platform integration is active the integration page will now have a button to reset mappings from Delinea Platform.
- Improvement: AD Privilege Password changer now has Remote Password Change timeout minutes Advanced Setting.
- Improvement: Better handling of unexpected heartbeat behavior to mitigate reported Distributed Engine stalling.
- Improvement: Connect As Credentials on Secret works better with SSH Keys for su user switching.
- Improvement: Updated links on the Security Hardening Report to new UI pages
- Improvement: When creating a new send to syslog task you no longer get a default schedule. Most of the templates didn't create a schedule, now they're all consistent.
- Improvement: Session monitoring search now supports searching by a single secret.
- Improvement: When a Secret is assigned to a site the user does not have access to due to Teams restriction, they will see the word "Restricted" instead of "Site Name (Inactive)".
- Improvement: Mitigated issue in large bulk secret actions.
- Improvements: The "Synchronization Running" message for DR will now only appear if there is a recorded start time for DR in the past and a finish time that is in the future.
- Improvement: Added Secret Field validation on the Template level to ensure users cannot create a "Secret Name" field on a template.
- Improvement: Default values for Secret Fields such as port will now be replicated for Disaster Recovery.
- Improvement: A user with only direct access to a report and the "browse reports" role permission can add that report to the dashboard.
- Improvement: The breadcrumbs within the RPC administration pages have been standardized. The links within Platform Vault Configuration Overview no longer cause the page to reload.
- Improvement: Report column preferences will be saved and applied when viewing a report.
- Improvement: The Secrets grid now updates displayed data and selected columns simultaneously.
- Improvement: Improved error logging and efficiency for calls coming from Delinea Platform.
- Improvement: Quick access filters now both apply when updated.
- Improvement: Knowledge base links within Platform Vault now link to their intended location.
- Improvement: Corrected edge case that could result in a session view audit being placed on the incorrect Secret.
- Improvement: The Parent Scan Template will be filtered to the type and will default to the first item in the list on create. The proper fields will be shown based on the type.
- Improvement: If a secret is inactivated after initially viewing the secret, a user that cannot view inactive secrets will no longer get an error from secret heartbeat.

- Improvement: Clicking cancel when editing folder permissions will clear any active filters.
- Improvement: Editing folder permissions now has a split button that allows for directly entering edit or add group/user mode.
- Improvement: The Secrets Quick Access link when collapsed, now targets the correct destination.
- Improvement: The Platform Opt In modal styling has been adjusted to no longer display with scroll bars.
- Improvement: Secret Share and Folder Permissions: Show disabled edit button until filters are loaded since split button does not yet support disabled.
- Improvement: API calls to /v[1/2]/secrets/{id} now update the Recents secrets data source.
- Improvement: When viewing Event Pipeline Activity details, selecting an Activity Detail record from the grid now displays the selected Activity's details.
- Improvement: Added query parameter for PipelineId to pass back when viewing specific pipeline activity
- Improvement: Minimum Heartbeat interval reduced from 15 to 5 minutes.
- Improvement: Discovery Scanner will not allow deletion until Secret selection is changed.
- Improvement: Remote Password Changing: Check for DNS Mismatch now visible and functional in Cloud.
- Improvement: EventTime token is available in pipeline scripts. $EventTime - event date and time of the event ("yyyy'-'MM'-'dd'T'HH':'mm':'ss").
- Improvement: The preview chips for Multifactor on Secrets have been removed.
- Improvement: Creating a User SSH Key in Platform downloads the private key with a proper filename.
- Improvement: Cipher Suite Configuration now allows configuration of allowed Host Key Algorithms.

Bug Fixes

- Fixed an issue where Viewing Session Connector Custom Launchers without access to the RDS Credentials secret would show an error.
- Fixed an issue where unplayable session recording videos would display an infinite load instead of the appropriate error.
- Fixed an issue where a secret template could be saved without RPC mappings configured.
- Fixed an issue where Pause times for ODBC Remote Password Changers were not adhered to. If you feel your RPC's are running slowly, check the pause times and remove them if they are not needed for the RPC action.
- Fixed an issue where Web Password Filler didn't work in certain instances due to an ambiguity in interpreting the Secret Server URL.
- Fixed an issue where setting custom expiration dates in all time zones did not work correctly.
- Fixed an issue where all event subscriptions did not fire for secrets in subfolders of the target folder.
- Fixed an issue where the secret name would incorrectly display on the New Discovery Import Rules page.
- Fixed an issue with negative numbers exporting incorrectly when exporting to a CSV file.
- Fixed an issue where a large number of SSH terminal connection history records causing timeouts.

- Fixed an issue with hidden days until deletion field when enabling deletion in the retention schedule. Added localization to error when trying to submit days less than or equal to the archive retention value.
- Fixed an issue with passwords being uneditable if RPC is set to use a Privileged Secret to which the user has no access to. Restored explanatory banner.
- Fixed an issue where secrets aren't synced with DevOps in cloud with when triggered by pipelines.
- Fixed issue in discovery where computer scans were sometimes throwing string truncation exceptions.
- Fixed an issue where TOTP Secret Settings edit button was available to users who could not edit the settings.
- Fix an issue with editing Session Connector Custom Launcher Port.
- Fixed a UI issue with the launcher popup window showing an option the user didn't have permission for.
- Fixed an issue where configuring a new session connector launcher might not show all available launcher types.
- Fixed an issue where configuring "Use Additional Prompt" on launchers might prevent save.
- Fixed an issue with the TemplateCreateSecret role link.
- Fixed an issue with View Launcher Password.
- Fixed an issue where users with only 'View' access on a Secret would be unable to view the Password if there was a custom launcher with arguments configured for that Secret Template.
- Fixed an issue with DSV sync for secret with file type fields and no file set.
- Fixed an issue with localization on folder Metadata page.
- Fixed an issue with sorting for Checkout User Id and Checkout User.
- Fixed an issue with ODBC password changing that broke postgres and mySQL changing.
- Fixed a logging issue with Dependency changes ran through Distributed Engine being skipped due to conditions.
- Fixed an issue where the generate SSH key returns a 500 exception.
- Fixed an issue where the SSHCipherSuiteModel GetAsync returned a 500 exception.
- Fixed an issue where the CreatePublicSSHKey returned a 500 Exception.
- Fixed an issue where Discovery Scanners could not be removed until the associated secrets had been edited.

Q

Secret Server: 11.5.000002 Release Notes

> This release resolves an issue that was discovered shortly after we published the 11.5.000001 General Availability release. The 11.5.000001 release was rolled back, and this release supersedes it.

> For convenience, this note also contains changes and features from the "Secret Server: 11.5.000000 EA Release Notes" on page 1657 and "Secret Server: 11.5.000001 GA Release Notes" on page 1654 release notes, not just the changes from 11.5.000002.

Release Dates and Notes

On-Premises: July 19, 2023.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.8.0

> The minimum required engine version is 8.3.0.0.

Protocol Handler: 6.0.3.26

New Features

Checked Out Secret View

We added checked out secret view to Quick Access in the Secrets Folder panel. This is a quick view, showing users all of the secrets that are currently checked out to them.

RADIUS Silent Answer

Silent answer is a new configuration option for RADIUS that allows setting the RADIUS response to a defined string value. This is to support push notification and other interactive variations in advanced RADIUS authentication configuration. The new setting replaces "Attempt User Password" and allows for sending the user password or another predefined string.

Check Out Recovery

We adjusted the behavior of "Force Check In" to allow secret owners with the "Force Check In" role permission to force check in secrets that are set to "Change Password on Check in." The secret is automatically checkout to the owner who initiate the force check in. This helps in situations where a checked out secret with a failing RPC configuration will not check in and remains with a user who cannot remediate the issue. With this change, an owner can take ownership of the checkout session, remediate the secret configuration, and then complete a normal secret check in.

Syslog Metadata for Launched Sessions

For built in launchers on a launch event, the launch target host is included within the details of the Syslog message as an additional "Host" field. Previously, this was only sent for launchers requiring host selection but now includes launchers with a static host-target mapping.

Launcher Administration Page Conversion

We updated the Launcher Administration pages under Secret Templates to use our new UI patterns with a modern design. No functionality is affected, but the page is more responsive and intuitive.

SSH Key Authentication Passphrase Requirement

We added a new configuration setting to the login configuration page that allows administrators to enable a mandatory requirement for passphrases when users generate SSH keys for SSH Terminal key authentication.

Enhancements

- Improvement: Upgraded error logging and efficiency for calls coming from Delinea Platform.
- Improvement: A user with only direct access to a report and the "browse reports" role permission can add that report to the dashboard.
- Improvement: Added a "Managed" field to the Discovery Network view to show when a discovery item is managed.
- Improvement: Added a Quick Access link to see all secrets you currently have checked out.
- Improvement: Added a refresh button to the Discovery Network view to refresh the data without having to refresh the entire page, losing the selected filtering.
- Improvement: Added a WMI Service Timeout setting to the cloud advanced configuration page to help with dependency changes that take more time than the allotted 60 seconds.
- Improvement: Added new tab to Discovery with overall metrics of discovered items and their statuses.
- Improvement: Added two columns to the Secret Grid—Checked Out User Id and Checked Out User. These show who has the secret checked out if the secret has check out enabled.
- Improvement: Added validation messages to password requirement rules for when password requirements are too complex to reliably generate a password.
- Improvement: Clicking on graph elements in discovery analysis now links to a filtered network view.
- Improvement: Disaster recovery date replication now syncs all SecretFieldLauncher items each time instead of just the updated ones.
- Improvement: Discovery Service Accounts Detail Page now shows services that run as the directory account as well as the computers on which that service runs
- Improvement: Enhanced the User Audit report to also exclude manually changed passwords.
- Improvement: Group membership assignment UI updated.
- Improvement: Group role assignment UI updated.
- Improvement: Implemented "select all" for the Discovery Network View.

- Improvement: Recently viewed secrets are now tracked within Platform. Configuration settings are now refreshed via navigation within Vault in Platform.
- Improvement: RPC heartbeat logs are now combined into a tabbed view with run buttons.
- Improvement: The breadcrumbs within the RPC administration pages have been standardized. The links within Platform Vault Configuration Overview no longer cause the page to reload.
- Improvement: The built-in "Everyone" group was renamed "All Vault Users."
- Improvement: The Delete folders function in disaster recovery can now delete more than 2100 folders or subfolders on the replica.
- Improvement: The Delinea Platform integration configuration now has additional validations for Login URL.
- Improvement: The PowerShell script timeout no longer defaults to 90 seconds. Instead, it now uses the value from the Event Pipelines Maximum Script Run Time (Minutes) setting in advanced configuration.
- Improvement: Updated Createuser.aspx to redirect to the new user creation page.
- Improvement: Updated group membership management pages to use new design patterns.
- Improvement: Updated the folder permission assignment UI.
- Improvement: Updated the group role assignment UI.
- Improvement: Updated the text and product descriptions used during the Platform opt-In experience.
- Improvement: Added a Managed field to the Discovery Network view to show when a discovery item is managed.
- Improvement: Added a Password Age column for display on the reworked Discovery Network View
- Improvement: Added a Quick Access link to see all Secrets you currently have checked out.
- Improvement: Added filters to the secret search API endpoint to filter the results by checked out status: paging.filter.showSecretsCheckedOutByUser and paging.filter.showCheckedOutSecrets
- Improvement: Added info to logs to indicate why users cannot match or create users in SSC. Find this at Secrets > Admin > Platform Integration > Logs tab. Common notifications include DuplicateUserMappedToDifferentProviderName: The user was initially setup to a different Platform source, the URL or userid (provider key) changed, indicating the original use was deleted. MaxLicensedUsersException: All licenses are taken so additional users cannot be added.
- Improvement: Added integration support for Platform users matching local SS users that do not have an @ in their name. If platform user is username@local or username@tenantname then the username portion will be used to match local users on the SS side.
- Improvement: Added support for LDAP RFC2307 group membership, used in OpenLDAP.
- Improvement: Added the option to require a passphrase for user public SSH keys.
- Improvement: Added validation messages to password requirement rules for when password requirements are too complex to reliably generate a password.
- Improvement: Discovery service accounts detail page now shows services that run as the directory account as well as the computers on which that service runs
- Improvement: Distributed engines no longer need directory services enabled to perform discovery.

- Improvement: Introduced a new Launch Secret role permission, which is needed to use launchers. This permission is automatically granted to roles with the View Secret permission, which previously controlled this behavior.
- Improvement: Removed the secretitemvaluetransitionhistory.aspx page and replaced it with an API endpoint, removing the possibility of bypassing the Hide Launcher Password control.
- Improvement: RPC heartbeat and password change logs are now full screen instead of a dialog box.
- Improvement: The PowerShell script timeout no longer defaults to 90 seconds. Instead, it now uses the value from the Event Pipelines Maximum Script Run Time (Minutes) setting in advanced configuration.
- Improvement: The new folder icon in the secret panel no longer shows if the user does not have the Administer Folders role permission.
- Improvement: The user audit report now has a filter panel and a description for how rotated secrets are calculated for this report.
- Improvement: There is now a pending RPC screen and a timer that checks you back in, blocking seeing secret info indefinitely.
- Improvement: Users can no longer access secrets that have failed processing a password change. Instead, they are shown a message stating the change failed.
- Improvement: We now initially load 60 secrets when viewing a grid to support 4k monitors. This was previously 30.
- Improvement: Within the details of the syslog message, there is now a username field that contains the mapped username for the launcher on a launch event. It appears as Username: [<username>] for the built in launchers.
- Improvement: Within the details of the syslog message, there is now a Host field with the value of the mapped host for the launcher on a launch event. It appears as Host: [<host>] for the built in launchers.

Bug Fixes

- Addressed an issue where users with only "view" access on a secret were unable to view the password if there was a custom launcher with arguments configured for that secret template.
- Fixed an issue where "Close launcher on check in" on the replication source would prevent sessions from being launched on the replica.
- Fixed an issue where a secret template could be saved without RPC mappings configured.
- Fixed an issue where all event subscriptions did not fire for secrets in subfolders of the target folder.
- Fixed an issue where discovery import would incorrectly assign a secret as an associated account, as opposed to a privileged account.
- Fixed an issue where existence of secrets set to expire over a hundred years in the future would cause expiration reports and event subscriptions to stop triggering.
- Fixed an issue where heartbeat and RPC log downloads would save without an extension. Now correctly saves with a .csv extension.
- Fixed an issue where launching secrets with URL List and session recording enabled displayed a "Bad Request" message.

- Fixed an issue where OpenID Connect or SAML accounts could not export secrets as they did not have a password and were not licensed for doublelock. DoubleLock is now available in professional licenses.
- Fixed an issue where the API endpoint api/v1/secrets/{id}/fields/{slug}/ logged an audit that the password was displayed when the actual password was not returned to the user due to "hide launcher password" being enabled. This could happen from some UI actions.
- Fixed an issue where the Edit button for User Management => Groups appeared when you did not have "Administer Role Assignment" permission. The action was denied in the API, so this was a cosmetic change.
- Fixed an issue where the folder audit page would unexpectedly show an access denied message.
- Fixed an issue where the folder permissions tab would load slowly with large numbers of users.
- Fixed an issue where the German localization for "Password Should Exclude" was incorrect.
- Fixed an issue where the new Discovery Network View UI would display a license error in the Professional Edition.
- Fixed an issue where secret name would incorrectly display on the New Discovery Import Rules page.
- Fixed an issue where the SubscriptionName condition for a notification rule would display the event subscription ID instead. It now correctly uses the name when the user has the appropriate roles to list the subscriptions.
- Fixed an issue where TOTP Secret Settings edit button was available to users who could not edit the settings.
- Fixed an issue where unplayable session recording videos would display an infinite load instead of the appropriate error.
- Fixed an issue with negative numbers exporting incorrectly when exporting to a CSV file.
- Fixed an issue with pinning a folder returned a "Folder not Found" error.
- Fixed an issue with secret search producing SQL errors for customers with a lot of secret templates.
- Fixed an issue where the SSH Proxy would stop processing new incoming connections.
- Fixed conditions that prevented users from being removed from a group due to the system incorrectly identifying that they would be unable to complete the same operation.
- Fixed issues with user and group syncing between Secret Server Cloud and the Delinea Platform.
- Fixed the TemplateCreateSecret role link.
- Updated links on the Security Hardening Report to new UI pages.
- Fixed an issue to improve Platform integration user sync if duplicate usernames were already in Secret Server.
- Fixed an issue where a secret template could be saved without RPC mappings configured.
- Fixed an issue where all event subscriptions did not fire for secrets in subfolders of the target folder.
- Fixed an issue where DR email alerts were not sent out.
- Fixed an issue where extended fields were not properly exported to CSV files.
- Fixed an issue where keystroke data from the advanced session recording agent did not appear in the keystroke activity details area of the playback page.

- Fixed an issue where large messages from distributed engines to engine workers would not process. Engine workers may have crashed especially frequently in environments having four or more workers, including Secret Server Cloud.
- Fixed an issue where LDAP sync via distributed engines would not work when the base DN was different from DC.
- Fixed an issue where links on the Session Monitoring page while in grid mode would not correctly link to Secret Server Cloud with authentication.
- Fixed an issue where the API endpoint api/v1/secrets/{id}/fields/{slug}/ logged an audit that the password was displayed when the actual password was not returned to the user due to hide launcher password being enabled.
- Fixed an issue where the Confirm Action button in the bulk operation dialog box would remain active while the operation is processing. This is now correctly disabled to prevent initiating the action multiple times.
- Fixed an issue where the SubscriptionName condition for a notification rule would display the event subscription ID instead. It now correctly uses the name when the user has the appropriate roles to list the subscriptions.
- Fixed an issue where the terminate session mouseover tooltip displayed incorrect text.
- Fixed an issue with a secret template name validation message not showing.
- Fixed an issue with negative numbers exporting incorrectly when exporting a CSV.
- Fixed an issue with new Platform trials not creating personal folders in Secret Server.
- Fixed an issue with stacked dialog boxes. The CSS styles for the Platform Opt In dialog box have been adjusted to align with Angular15.
- Fixed conditions that prevented users from being removed from a group due to the system incorrectly identifying that they would be unable to complete the same operation.
- Fixed issues with user and group syncing between Secret Server Cloud and Platform.
- Fixed usability on specific UI areas for a better user experience.
- Updated Createuser.aspx to redirect to the new user management.

Future and Recent Deprecations

This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server: 11.5.000001 GA Release Notes

This release has been temporarily delayed while we while we investigate a newly reported issue.

This note only contains changes between the EA and GA releases. Earlier changes and all the features are in the "Secret Server: 11.5.000000 EA Release Notes" on page 1657).

Release Dates and Notes

On-Premises: July 6, 2023.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.8.0

> The minimum required engine version is 8.3.0.0.

Protocol Handler: 6.0.3.26

New Features

See the "Secret Server: 11.5.000000 EA Release Notes" on page 1657) for features.

Enhancements

- Improvement: A user with only direct access to a report and the "browse reports" role permission can add that report to the dashboard.
- Improvement: Added a "Managed" field to the Discovery Network view to show when a discovery item is managed.
- Improvement: Added a Quick Access link to see all secrets you currently have checked out.
- Improvement: Added a refresh button to the Discovery Network view to refresh the data without having to refresh the entire page, losing the selected filtering.
- Improvement: Added a WMI Service Timeout setting to the cloud advanced configuration page to help with dependency changes that take more time than the allotted 60 seconds.
- Improvement: Added new tab to Discovery with overall metrics of discovered items and their statuses.
- Improvement: Added two columns to the Secret Grid—Checked Out User Id and Checked Out User. These show who has the secret checked out if the secret has check out enabled.
- Improvement: Added validation messages to password requirement rules for when password requirements are too complex to reliably generate a password.
- Improvement: Clicking on graph elements in discovery analysis now links to a filtered network view.
- Improvement: Disaster recovery date replication now syncs all SecretFieldLauncher items each time instead of just the updated ones.
- Improvement: Discovery Service Accounts Detail Page now shows services that run as the directory account as well as the computers on which that service runs
- Improvement: Enhanced the User Audit report to also exclude manually changed passwords.
- Improvement: Group membership assignment UI updated.
- Improvement: Group role assignment UI updated.
- Improvement: Implemented "select all" for the Discovery Network View.
- Improvement: Recently viewed secrets are now tracked within Platform. Configuration settings are now refreshed via navigation within Vault in Platform.
- Improvement: RPC heartbeat logs are now combined into a tabbed view with run buttons.

- Improvement: The breadcrumbs within the RPC administration pages have been standardized. The links within Platform Vault Configuration Overview no longer cause the page to reload.
- Improvement: The built-in "Everyone" group was renamed "All Vault Users."
- Improvement: The Delete folders function in disaster recovery can now delete more than 2100 folders or subfolders on the replica.
- Improvement: The Delinea Platform integration configuration now has additional validations for Login URL.
- Improvement: The PowerShell script timeout no longer defaults to 90 seconds. Instead, it now uses the value from the Event Pipelines Maximum Script Run Time (Minutes) setting in advanced configuration.
- Improvement: Updated Createuser.aspx to redirect to the new user creation page.
- Improvement: Updated group membership management pages to use new design patterns.
- Improvement: Updated the folder permission assignment UI.
- Improvement: Updated the group role assignment UI.
- Improvement: Updated the text and product descriptions used during the Platform opt-In experience.

Bug Fixes

- Fixed an issue where "Close launcher on check in" on the replication source would prevent sessions from being launched on the replica.
- Fixed an issue where a secret template could be saved without RPC mappings configured.
- Fixed an issue where all event subscriptions did not fire for secrets in subfolders of the target folder.
- Fixed an issue where discovery import would incorrectly assign a secret as an associated account, as opposed to a privileged account.
- Fixed an issue where existence of secrets set to expire over a hundred years in the future would cause expiration reports and event subscriptions to stop triggering.
- Fixed an issue where heartbeat and RPC log downloads would save without an extension. Now correctly saves with a .csv extension.
- Fixed an issue where launching secrets with URL List and session recording enabled displayed a "Bad Request" message.
- Fixed an issue where OpenID Connect or SAML accounts could not export secrets as they did not have a password and were not licensed for doublelock. DoubleLock is now available in professional licenses.
- Fixed an issue where the API endpoint api/v1/secrets/{id}/fields/{slug}/ logged an audit that the password was displayed when the actual password was not returned to the user due to "hide launcher password" being enabled. This could happen from some UI actions.
- Fixed an issue where the Edit button for User Management => Groups appeared when you did not have "Administer Role Assignment" permission. The action was denied in the API, so this was a cosmetic change.
- Fixed an issue where the folder audit page would unexpectedly show an access denied message.
- Fixed an issue where the folder permissions tab would load slowly with large numbers of users.
- Fixed an issue where the German localization for "Password Should Exclude" was incorrect.

- Fixed an issue where the new Discovery Network View UI would display a license error in the Professional Edition.
- Fixed an issue where secret name would incorrectly display on the New Discovery Import Rules page.
- Fixed an issue where the SubscriptionName condition for a notification rule would display the event subscription ID instead. It now correctly uses the name when the user has the appropriate roles to list the subscriptions.
- Fixed an issue where TOTP Secret Settings edit button was available to users who could not edit the settings.
- Fixed an issue where unplayable session recording videos would display an infinite load instead of the appropriate error.
- Fixed an issue with negative numbers exporting incorrectly when exporting to a CSV file.
- Fixed an issue with pinning a folder returned a "Folder not Found" error.
- Fixed an issue with secret search producing SQL errors for customers with a lot of secret templates.
- Fixed an issue where the SSH Proxy would stop processing new incoming connections.
- Fixed conditions that prevented users from being removed from a group due to the system incorrectly identifying that they would be unable to complete the same operation.
- Fixed issues with user and group syncing between Secret Server Cloud and the Delinea Platform.
- Fixed the TemplateCreateSecret role link.
- Updated links on the Security Hardening Report to new UI pages.

Future and Recent Deprecations

This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server: 11.5.000000 EA Release Notes

Release Dates and Notes

On-Premises: June 9, 2023.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.8.0

Protocol Handler: 6.0.3.26

New Features

Checked Out Secret View

We added checked out secret view to Quick Access in the Secrets Folder panel. This is a quick view, showing users all of the secrets that are currently checked out to them.

RADIUS Silent Answer

Silent answer is a new configuration option for RADIUS that allows setting the RADIUS response to a defined string value. This is to support push notification and other interactive variations in advanced RADIUS authentication configuration. The new setting replaces "Attempt User Password" and allows for sending the user password or another predefined string.

Check Out Recovery

We adjusted the behavior of "Force Check In" to allow secret owners with the "Force Check In" role permission to force check in secrets that are set to "Change Password on Check in." The secret is automatically checkout to the owner who initiate the force check in. This helps in situations where a checked out secret with a failing RPC configuration will not check in and remains with a user who cannot remediate the issue. With this change, an owner can take ownership of the checkout session, remediate the secret configuration, and then complete a normal secret check in.

Syslog Metadata for Launched Sessions

For built in launchers on a launch event, the launch target host is included within the details of the Syslog message as an additional "Host" field. Previously, this was only sent for launchers requiring host selection but now includes launchers with a static host-target mapping.

Launcher Administration Page Conversion

We updated the Launcher Administration pages under Secret Templates to use our new UI patterns with a modern design. No functionality is affected, but the page is more responsive and intuitive.

SSH Key Authentication Passphrase Requirement

We added a new configuration setting to the login configuration page that allows administrators to enable a mandatory requirement for passphrases when users generate SSH keys for SSH Terminal key authentication.

Enhancements

- Improvement: Added a Managed field to the Discovery Network view to show when a discovery item is managed.
- Improvement: Added a Password Age column for display on the reworked Discovery Network View
- Improvement: Added a Quick Access link to see all Secrets you currently have checked out.
- Improvement: Added filters to the secret search API endpoint to filter the results by checked out status: paging.filter.showSecretsCheckedOutByUser and paging.filter.showCheckedOutSecrets
- Improvement: Added info to logs to indicate why users cannot match or create users in SSC. Find this at Secrets > Admin > Platform Integration > Logs tab. Common notifications include DuplicateUserMappedToDifferentProviderName: The user was initially setup to a different Platform source, the URL or userid (provider key) changed, indicating the original use was deleted. MaxLicensedUsersException: All licenses are taken so additional users cannot be added.

- Improvement: Added integration support for Platform users matching local SS users that do not have an @ in their name. If platform user is username@local or username@tenantname then the username portion will be used to match local users on the SS side.
- Improvement: Added support for LDAP RFC2307 group membership, used in OpenLDAP.
- Improvement: Added the option to require a passphrase for user public SSH keys.
- Improvement: Added validation messages to password requirement rules for when password requirements are too complex to reliably generate a password.
- Improvement: Discovery service accounts detail page now shows services that run as the directory account as well as the computers on which that service runs
- Improvement: Distributed engines no longer need directory services enabled to perform discovery.
- Improvement: Introduced a new Launch Secret role permission, which is needed to use launchers. This permission is automatically granted to roles with the View Secret permission, which previously controlled this behavior.
- Improvement: Removed the secretitemvaluetransitionhistory.aspx page and replaced it with an API endpoint, removing the possibility of bypassing the Hide Launcher Password control.
- Improvement: RPC heartbeat and password change logs are now full screen instead of a dialog box.
- Improvement: The PowerShell script timeout no longer defaults to 90 seconds. Instead, it now uses the value from the Event Pipelines Maximum Script Run Time (Minutes) setting in advanced configuration.
- Improvement: The new folder icon in the secret panel no longer shows if the user does not have the Administer Folders role permission.
- Improvement: The user audit report now has a filter panel and a description for how rotated secrets are calculated for this report.
- Improvement: There is now a pending RPC screen and a timer that checks you back in, blocking seeing secret info indefinitely.
- Improvement: Users can no longer access secrets that have failed processing a password change. Instead, they are shown a message stating the change failed.
- Improvement: We now initially load 60 secrets when viewing a grid to support 4k monitors. This was previously 30.
- Improvement: Within the details of the syslog message, there is now a username field that contains the mapped username for the launcher on a launch event. It appears as Username: [<username>] for the built in launchers.
- Improvement: Within the details of the syslog message, there is now a Host field with the value of the mapped host for the launcher on a launch event. It appears as Host: [<host>] for the built in launchers.

Bug Fixes

- Fixed an issue to improve Platform integration user sync if duplicate usernames were already in Secret Server.
- Fixed an issue where a secret template could be saved without RPC mappings configured.
- Fixed an issue where all event subscriptions did not fire for secrets in subfolders of the target folder.
- Fixed an issue where DR email alerts were not sent out.

- Fixed an issue where extended fields were not properly exported to CSV files.
- Fixed an issue where keystroke data from the advanced session recording agent did not appear in the keystroke activity details area of the playback page.
- Fixed an issue where large messages from distributed engines to engine workers would not process. Engine workers may have crashed especially frequently in environments having four or more workers, including Secret Server Cloud.
- Fixed an issue where LDAP sync via distributed engines would not work when the base DN was different from DC.
- Fixed an issue where links on the Session Monitoring page while in grid mode would not correctly link to Secret Server Cloud with authentication.
- Fixed an issue where the API endpoint api/v1/secrets/{id}/fields/{slug}/ logged an audit that the password was displayed when the actual password was not returned to the user due to hide launcher password being enabled.
- Fixed an issue where the Confirm Action button in the bulk operation dialog box would remain active while the operation is processing. This is now correctly disabled to prevent initiating the action multiple times.
- Fixed an issue where the SubscriptionName condition for a notification rule would display the event subscription ID instead. It now correctly uses the name when the user has the appropriate roles to list the subscriptions.
- Fixed an issue where the terminate session mouseover tooltip displayed incorrect text.
- Fixed an issue with a secret template name validation message not showing.
- Fixed an issue with negative numbers exporting incorrectly when exporting a CSV.
- Fixed an issue with new Platform trials not creating personal folders in Secret Server.
- Fixed an issue with secret search producing SQL errors for customers with a lot of secret templates.
- Fixed an issue with stacked dialog boxes. The CSS styles for the Platform Opt In dialog box have been adjusted to align with Angular15.
- Fixed conditions that prevented users from being removed from a group due to the system incorrectly identifying that they would be unable to complete the same operation.
- Fixed issues with user and group syncing between Secret Server Cloud and Platform.
- Fixed usability on specific UI areas for a better user experience.
- Updated Createuser.aspx to redirect to the new user management.

Future and Recent Deprecations

This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server: 11.4.000031 Release Notes

11.4.000032 is a retroactive patch for this release. Please see "Secret Server 11.7.000001 Release Notes" on page 1596 for details.

Release Dates and Notes

On-Premises: May 17, 2023.

This maintenance release fixes the issues that caused us to roll back release 11.4.000030 to 11.4.000002 after discovering some bugs that could significantly affect a minority of customers.

Please see "Secret Server: 11.4.000030 Release Notes" below for complete release notes.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.7.0

Protocol Handler: 6.0.3.26

Bug Fixes

- Fixed an issue where the engine worker process would fail to receive and process large messages.
- Fixed an issue where the engine worker process would crash when proxy session recording was enabled in multi-node environments.

Future and Recent Deprecations

This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server: 11.4.000030 Release Notes

Release Dates and Notes

On-Premises: May 2, 2023.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.7.0

Protocol Handler: 6.0.3.26

Important: We rolled back release 11.4.000030 to 11.4.000002 after discovering some bugs that can significantly affect a minority of customers. See the first two Known Issues for details. We apologize for any inconvenience, especially for the delay in releasing the new, resilient secrets and other enhancements. It is a top priority for engineering to fix this. An updated release should be available by May 20th, 2023.

Known Issues

- Very large messages sent from distributed engines back to Secret Server are not processed due to a bug in the message processing code. This can include messages related to Active Directory synchronization, discovery, and keystroke recording. See the note at the top of this topic.
- In installations with multiple nodes, launched sessions that are RDP or SSH proxied and recorded may cause the engine worker on a node to crash. The probability of a crash increases with each additional node running,

becoming nearly certain at eight or more nodes. See the note at the top of this topic.

- The distributed engine (DE) package that came with Secret Server (Cloud and On-Premise) 11.3.x prevents Secret Server from performing a DE auto-upgrade—making a manual upgrade necessary. See the Distributed Engine Auto-Upgrade Does Not Work bulletin for details.

  Note: This is not required for admins who already completed a manual upgrade for any version of 11.4.

Enhancements

Secret Folder Panel Redesign

We reworked the secret folder panel for additional functionality and a more streamlined user experience. Direct access to the folder panel from outside of the secrets view was found to unnecessarily clutter the navigation menu, and the panel is now visible on the Secrets page. In addition, we enhanced the following:

- We added a "Quick Access" section to the folder panel, which offers a single page combining the following sections:
  - Search
  - Favorite Secrets
  - Most Used Secrets
  - Session Secrets—Secrets accessed this session, allowing the user to return to a secret they have accessed in this browser session
  - Recent Secrets—Most recently accessed secrets, within this session or others
  - Shared With Me—Secrets that are shared with you but not in folders that you may view
- Added a "New Folder" button to the top of the folder tree.
- Pinned folders are now placed at the top of the tree instead of listed in a dropdown. Pinned folders give easier access to your favorite folders. When a pinned folder is selected, the displayed folder tree is based on the that pinned folder rather than the entire tree. The same applies to the content of "Quick Access," which displays secrets in the selected, pinned folder.
- A guide now displays for new users the first time they view the secrets page, introducing components and changes.

Other Enhancements

- DR: Created more robust data ambiguity handling when data replication processes a table where there is a multi-field unique key. These included giving precedence to source data when applicable or throwing an error when an ambiguity cannot be resolved.
- DR: Added an advanced configuration setting (defaulting to 3 hours) so that a long-running DR process will detect the configured amount of elapsed time and end the DR process, forcing the business user to manually run it again.
- DR: Read-only mode can now be enabled in Secret Server Cloud on the disaster recovery configuration page.
- Added a discovery import rule to the new network viewer.

- Added a link to the public SSH keys, when enabled, on both the user preference page and the administration tools section.
- Added a knowledge base link for Platform regions as part of the Platform Optin Experience.
- Added support for LDAP RFC2307 group membership, used in OpenLDAP.
- Discovery rules and dependencies grid can now be filtered by discovery source. Rule grid now also has discovery source available as a column.
- Local Admin column added to new discovery network view.
- Secret template name on the secret general tab is now a direct link to the template.
- The "Send Test Email" button now functions in read only mode.
- The report CSV download is now encoded so that certain Turkish characters appear, and the mime type was changed to text/csv.
- The unlimited admin page in configuration preview now has a link to open the unlimited admin audit.
- HSM Integration—RSA OAEP Padding Support. We added OAEP padding as a new configurable option when enabling or rotating the HSM integration. This is in anticipation of the planned deprecation of PKCS padding by NIST. Current configurations are unaffected, but this option is available when rotating the HSM key, or configuring a new integration.

Bug Fixes

Disaster Recovery

- Fixed an issue where DR email alerts were not being sent out.
- Fixed issues for password-requirement character-set data replication in the DR feature.
- Fixed an issue with disaster recovery replication where replicated custom launchers were not visible on their associated secrets.
- Fixed an issue with the disaster recovery logging process so that only error-free data replications are marked as successful.
- Fixed issue when replicating data for disaster recovery where pre-existing users on the replica that do not exist on the source could lose their All Vault Users group membership.
- Fixed replication to allow duplicate names to be replicated individually during disaster recovery. Groups with the same name will still be consolidated during replication when they share values for AD Guid, IsPersonal, IsPlatform, and DomainId.
- Password requirements are now replicated from source to replica as part of disaster recovery.

Other Bug Fixes

- Corrected logic that allowed password requirement consumers to bypass non-replicated secrets
- Dates in the report export no longer include the "Z" for UTC when server time is used and ISO date format is selected because the date/time is the server configured time and not necessarily UTC. That is, the date is ISO format and the server-configured time but does not include the offset. In some specific configurations when user

format was selected, the timezone offset would be applied based on the actual server timezone and not the configured timezone.

- Discovery-specific OUs now returns results when the page is initially loaded.
- Due to security reasons, we removed the GET endpoint for /secretserversettings/export and replacing it with a POST endpoint where we can transmit the password securely. The contents of the payload are the same, except for new "password" and "doubleLockPassword" fields, and the entire payload is contained within a parent "data" object.
- Fixed a bug that caused launcher session failure on secrets that were expired on checkout but then disabled checkout via policy. Also, retroactively fixed this situation on secrets.
- Fixed a string truncation error. Expanded the user setting size to resolve issue for some customers with lots of columns for a grid.
- Fixed a timing problem where secret favorites might not initialize if the secret grid loads very quickly.
- Fixed an CSS issue where clicking the "Browse all folders" button caused folder names to overlap.
- Fixed an issue that had platform logout redirecting to a different tenant.
- Fixed an issue where "Web Launcher requires Incognito Mode" was not being respected when enabled. Descriptive text added on web launcher mapping for restricting input fields.
- Fixed an issue where a bulk action was applied to all secrets when select all is checked but a template or folder filter was applied.
- Fixed an issue where a default error page was presented when accessing certain URLs as opposed to a more technical error.
- Fixed an issue where a dependency fails to work when moved to another group or order due to the run condition. When a secret dependency is updated to the first sort order or to a new group, the run condition is cleared. This addresses an issue where a secret dependency with a run condition would not run when it was the first secret dependency in its group.
- Fixed an issue where a Get Folders API call did not returns all descendants, breaking some customer integrations. To retrieve direct children only, use the new LimitToDirectDescendents parameter.
- Fixed an issue where an HSM could not be disabled.
- Fixed an issue where Azure domain accounts were unable to access Secret Server SSH Terminal with a public key. You can now log into Terminal with an Azure Active Directory account using SSH Key Integration. AAD logins to Terminal via password cannot be done.
- Fixed an issue where changing the time zone on the secret audit page did nothing and refreshing the page returned to the default time zone. When the server time zone is different, the time zone picker should show and the date column for audit should render in the selected time zone.
- Fixed an issue where connecting using an SSH key on another secret did not work with "SSH key only" secrets.
- Fixed an issue where data retention under PII removed monitored recordings or user audits related to monitored recordings. Data retention under database size management will still remove monitored recordings and related user audit records.

- Fixed an issue where duplicate user names were throwing an error. When logging in as a local user, we now ignore any Platform native users that may have the same login name, instead of erroring.
- Fixed an issue where event pipeline email notifications were not sent if the email task had an email template selected.
- Fixed an issue where exported computer scan logs were incomplete. Discovery logs now export more than 250 records.
- Fixed an issue where folders and sub-folders were missing secrets with UAM enabled. Left nav max folders default limit increased to 1,000. Setting dialog added to set the user preferred limit. Folder browser now loads 100 records at a time on scroll instead of just 30.
- Fixed an issue where GET /internals/secret-detail/{id}/launcher/{launchertypeid} threw an exception. We now show a friendly error message when launching a secret With Jumpbox Route with RDP that it is missing an SSH launcher
- Fixed an issue where OpenLDAP directory services group-search filter was not working.
- Fixed an issue where PowerShell dependency changer arguments were not being passed into the script.
- Fixed an issue where secret field data over a certain length may be rejected by the database upon replication.
- Fixed an issue where Secret PasswordComplianceCode was not updated after password field/PasswordReq change.
- Fixed an issue where secret template fields of type file no longer showed the drop down options when editing the field.
- Fixed an issue where session monitoring grid view showed the system and not the Secret Server. The secret session search date in the grid and card both now show in the selected time zone and the grid has the timezone picker when relevant.
- Fixed an issue where the folder list disappeared if UAM is enabled and when the "All Folders" toggle is selected in the sidebar. Folders in the tree will now be limited to only show 125 folders per tree. Once there are 125+ subfolders a "Browse all folders" option will appear in the folder tree. This link will take the user to a grid that only shows folders with a search. The grid has paging so it will load 30 folders at a time as the user scrolls. This will help support instances when users have thousands of subfolders. If there are more than 30 subfolders in a folder the secret grid will show a link to the new folder browser. This used to open a dialog to the folder tree which would also run into performance issues when users had over 1,000 subfolders.
- Fixed an issue where the folder tree disappeared when there were more than 1,000 folders accessed and UAM was enabled.
- Fixed an issue where the pipeline activity status stopped updating after the "Send to Email" task
- Fixed an issue where the Preserve Client SSH Process did not appear for process launchers
- Fixed an issue with heartbeat failures if a secret had checkout enabled.
- Fixed an issue with secret search would produce an excessively long URL that would sometimes throw an HTTP 404 error. The secret search API endpoint now accepts a filter param called ExtFieldsCombined, which is a comma delimited list of all extended fields to include in the results. This field is now used by the secret grid to help reduce the size of the URL when many secret fields are exposed for display to avoid the IIS 2k length restriction on GETs.

- Fixed an issue with SSH proxy "Tunnel RDP Connections" performance degradation (high CPU usage).
- Fixed an issue with the data retention page background color.
- Fixed discovery network view to ensure when searching you should be able to find all items under your current levels. However, when looking at a level you only see that level.
- Fixed issue with the password compliance report updating very slowly or not refreshing after either a template or direct PasswordRequirement password field change.
- Fixed issues related to RabbitMQ channel and queue growth and corruption-related issues due to connection interruption causing premature queue deletion.
- Fixed the default timestamp format for CEF.
- The "All Secrets" CSV download now correctly shows the folder name instead of the folder ID.
- The secret policy approvers "default only" option now displays correctly when updated.
- Updated the advanced session recording agent version label on the agent issues page to correctly state that it is the minimum required version, not the current version.
- Fixed an issue where a purge of inactive sessions longer than three minutes was occurring when the Sessions Monitoring page was displayed. It did not take into account the SSH proxy timeout. The page now obeys the timeouts.

Future and Recent Deprecations

Note: This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server: 11.4.000002 GA Release Notes

Release Dates and Notes

On-Premises: March 7, 2023

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.3.0

Protocol Handler: 6.0.3.26

Known Issues

The distributed engine (DE) package that came with Secret Server (Cloud and On-Premise) 11.3.x prevents Secret Server from performing a DE auto-upgrade—making a manual upgrade necessary. See the Distributed Engine Auto-Upgrade Does Not Work bulletin for details.

Changes Since the Early Availability Release

The following updates focus on improving the stability, performance, and functionality of disaster recovery replication, as well as resolving various issues and bugs that were identified. Additionally, updates were made to localizations and character sets to ensure compatibility and reliability.

The changes are

- The stability and performance of replication of Folder Permissions in disaster recovery has been increased.
- Secret Items from the source are combined with ones from the replica during disaster recovery replication when they have matching SecretIDs and SecretFieldIds.
- Disaster recovery now stops processing and disables replication when it encounters a fatal error and sends a notification of the error to all the disaster recovery admins.
- An issue with disaster recovery folder synchronization selection was resolved, and personal folders can now be selected for either allow or block lists.
- An issue with folder name collisions in disaster recovery synchronization was fixed.
- Localizations were updated.
- Older character sets that failed to replicate when running disaster recovery were fixed.

New Features

User Interface Streamlining: Classic UI Removed

The classic UI is no longer available as an option, and can no longer be enabled. This followed notifications of phased deprecations in prior releases. Several improvements have been made to the UI based on feedback from customers regarding this change.

Checkout Extension Maximum Limit

We created a global configuration setting that allows administrators to set a maximum secret-checkout extension interval. This provides additional admin control by specifying granular limitations to users extending a checked out secret. The time limitation begins at the point of checkout extension and extension time defaults to the set checkout time

Disaster Recovery Enhancements

"Replicated User Status on Disaster Recovery Source" configuration can now be set to:

- Mirror Source, which is the existing behavior
- Disabled by default

New Synchronization Items:

- Character sets
- AD users and groups

Discovery User Experience Improvements

We updated the discovery user experience to reflect the style and design of the application. The legacy pages are still available; however; the new interface items are ready for use, and we welcome feedback on these items. The legacy pages can be accessed by browsing to the relevant new interface and clicking the "View Legacy Page" button. The improvements are:

- Network view is available as a tab on the main Discovery Sources page.
- Network view displays the same results as the legacy page, in a single filterable grid, as opposed to individual tabs for different types of scanners. The new page has the functionality of the legacy page but in a more-responsive and updated design.
- There is a new filter menu, allowing extensive filtering options of this data.
- Each item in the network view list links to a details page allowing review of the discovered data, as opposed to being viewed inline.
- The grid data is exportable as a CSV, similar to other grids.
- Scanner, scan template, command set and search filter configuration are available from the Discovery Configuration Options > Scanner Definition button on the Discovery Configuration page.
- Source scanner configuration is available in the Discovery Source Configuration page as a tab.
- We extensively redesigned the scanner configuration UI to make this experience more intuitive, and scanners are now displayed in a workflow view.

Generated and Created Password Improvements

Password Complexity Indicator

There is a new visual indicator in the password complexity rules that provides the user with a better understanding of the strength of their password. The combined score considers both entropy score (brute force defense) and character limitations (social engineering defense). In the case that the score is deemed too low, the UI provides recommendations to the user on how to increase password strength.

New Password Rules

We introduced character rules to password complexity selection to enhance the strength of generated and created passwords, if enabled. The new rules provide flexibility in the granularity of the rules. Each selection impacts both entropy and overall strength score. The rules include minimum characters from:

- Lowercase letters
- Symbols
- Numerals
- Uppercase letters

Opt-In Engine Upgrades

Distributed engine upgrades are no longer mandatory for every release. We added a new setting to the Distributed Engine Configuration page to set the minimum required engine version. Modifying this will trigger an automatic update for any engine below this version.

In the action menu for an engine on the Sites page, a manual upgrade can be triggered for individual engines below the latest version, which prompts the engine to update when it next calls in.

When changes are made needing an upgrade, the minimum required version is updated during the update process, and all engines update immediately.

"Run Scripts" Role Permission

We created a new "Run Scripts" role permissions to separate privileges in script management. Holders of the "View Scripts" role permission cannot execute test runs of scripts, and the new role permission must be assigned to perform this task.

Administer Scripts remains unchanged and allows view, edit, and run permissions.

Syslog Timestamps

There is a new setting in Syslog configuration allowing the selection of timestamp formatting. The standard for Syslog indicates that ISO timestamps should be used; however, some consumers use the legacy format. There is now a selection between Syslog and ISO format. Syslog will be the default for upgrades to allow current configurations to retain their behavior, and ISO format is the default in new instances.

Site-Specific FIPS Configuration

Individual sites are now configurable for FIPS compatibility. The setting is available on the Administration > Distributed Engine > Site configuration page, in the Engine Default Settings dialog box. All engines on a site will use this setting, overriding the global setting, which is configured at Administration > Configuration > Security.

Enhancements

- Added a configuration option to disable the SMB heartbeat fallback check.
- Added a secret policy setting to control "Change Password Upon Check-In" behavior. Previously, this was automatically enabled if "Require Check Out" was enabled.
- Added additional debugging output for SSH proxy when using the "ALL" logging level.
- Added audits for emailing and downloading reports.
- Added endpoints for Update Password Type Auth, Get Password Type Auth, and Create Password Type Auth. These allow you to create and update records for the command arguments on RPC command set up.
- Added the configuration setting "Allow Files without Extension" to the configuration preview.
- Added the internal site connector configuration to the configuration preview.
- Added the User parameter to the IBM iSeries Mainframe connection for launching, password changing, and heartbeat.
- Bulk edit share now has a "None" permission which allows removing permissions.
- Changed IIS web.config configuration to disallow access to the file uploads folder.
- Enabled more connection classes to use read-only mode.
- Enhanced secret export logging.
- Improved performance of dependency matching within discovery.
- Improved performance of the role assignment page and added a user panel on the same page.
- Improved performance of the secret to computer matching operation that runs as part of discovery.

- In the data replication summary log now lists in alphabetical order, success and version number will appear before the list of items and any errors are appended at the end.
- Optimized application caching.
- Updated the new UI to allow newly generated SSH keys to have a blank passphrase, which matches legacy UI functionality.

Bug Fixes

- Fixed a memory leak in SSH proxy.
- Fixed a SAML audit error.
- Fixed an error that occurred when multiple identical domains were created.
- Fixed an error where the new SSH proxy custom SSH cipher suite settings were not picked up by distributed engines.
- Fixed an issue in the heartbeat status by day report that would cause the same secret to be counted twice on days where the secret transitions between heartbeat failure and success.
- Fixed an issue where "Days Until Expiration" value on the secrets grid would show a large negative number if expiration is forced. This now displays "Expiration forced."
- Fixed an issue where "requires approval type" could not be set by policy.
- Fixed an issue where a user with edit permissions could not rename a folder.
- Fixed an issue where an "invalid SQL error" was incorrectly displayed when a report timed out.
- Fixed an issue where an error was displayed in an edit field dialog box.
- Fixed an issue where an inbox notification was not clearing in Secret Server Cloud.
- Fixed an issue where completed master encryption key rotation would not show as such.
- Fixed an issue where converting a secret in a folder with a launcher settings policy threw an error.
- Fixed an issue where deleted Active Directory groups were not correctly marked as inactive when synchronized.
- Fixed an issue where disaster recovery would log many password requirement errors.
- Fixed an issue where failing Syslog/SIEM messages did not respect updated Syslog Server configuration.
- Fixed an issue where file contents during SFTP/SCP file transfers were included in the session keystroke recordings.
- Fixed an issue where installation on specific dates on servers with a dd/mm/YYYY localization configuration would prevent some configuration settings from being read.
- Fixed an issue where Local site could not be configured to use Custom SSH Cipher Suite settings when set to process on the Web Site.
- Fixed an issue where manual backup did not work in maintenance mode.
- Fixed an issue where master encryption key rotation would fail due to discovery import rules running at the same time.
- Fixed an issue where missing file attachments caused DR replications to fail.

- Fixed an issue where monitoring and termination of live sessions was not displayed in the UI. This now takes the user to the regular session playback page, which displays the live session.
- Fixed an issue where PowerShell-based dependency changers would not correctly pass arguments.
- Fixed an issue where PuTTY would close immediately following a session error. This was due to a default setting change in PuTTY, which is now explicitly set to remain open on installation or update of the protocol handler. This requires a protocol handler update.
- Fixed an issue where reports generate an application error if users navigated away from the report while it was loading.
- Fixed an issue where scrolling the secrets grid view would deselect items.
- Fixed an issue where Secret Server did not correctly react when two templates have the same field if one had spaces that the other did not.
- Fixed an issue where secrets would not open when users have folder view and secret list permissions. The secret audit within that folder should be accessible.
- Fixed an issue where session recording would sometimes show a 500 error, even though the client would retry. Replaced this with a HTTP 429 response, explicitly informing the client to retry.
- Fixed an issue where session recordings could not be saved to a UNC file path due to missing permissions on the root of the path.
- Fixed an issue where sorting by folder path in the secret grid view would return an error.
- Fixed an issue where SSH dependencies would not process on distributed engines.
- Fixed an issue where SSH Proxy would not allow a launcher to connect in maintenance mode. This is now possible in non-recorded sessions—recording is not possible in maintenance mode.
- Fixed an issue where the advanced session recording agent would attempt to make many reconnections in a short time span.
- Fixed an issue where the 64-bit protocol handler would not function when "Enable Protocol Handler Auto-Update" was enabled.
- Fixed an issue where the folder picker would not populate when adding folders in event pipeline policies while in unlimited admin mode.
- Fixed an issue where the IBM iSeries password changer was not properly adding the model value to the connection string.
- Fixed an issue where the light/dark mode toggle displayed "Enable Dark Mode" even though the UI was already in dark mode. This was due to a dark mode browser preference and no explicit user preference having been set.
- Fixed an issue where the SAML AuthnRequest was sending a blank RequestedAuthnContext when Authentication Context was set in the Identity Provider Configuration.
- Fixed an issue where the Secret Server website would not load if the internal site connector is unavailable at startup.
- Fixed an issue where the terminate option on the session playback page was missing.
- Fixed an issue where the test buttons would not function for the Oracle Account Ver. 2 password changer.

- Fixed an issue where the UI session monitoring search required additional permissions to load.
- Fixed an issue where the unlimited administrator watermark could block interaction with some page elements.
- Fixed an issue where the wrong error message would be shown when trying to apply an invalid data source key under data replication.
- Fixed an issue where user permissions on replica instances were removed erroneously when data replication ran.
- Fixed an issue with check in when the "Check In Secret on Launcher Close" and "Close Launcher on Check In Secret" settings were both false.
- Fixed an issue with DR replication where some operations would give an error "The incoming request has too many parameters."
- Fixed an issue with excessive CPU usage for RDPWin.exe. Protocol handler and session connector no longer track or record processes using WMI. Now they use native Windows calls, reducing CPU usage of the Windows WMI Provider. The exception is when "Run as secret credentials" is used—it still uses the WMI process tracking.
- Fixed an issue with replicating domain users. This now correctly links the user with the replicated domain.
- Fixed an issue with slow loading sessions failing to load when using session connector. This required an update to the latest protocol handler (RDS) on the session connector server.
- Fixed an issue with the group filter in event pipelines to ensure precise name matching is correctly used.
- Fixed an issue with the SearchSecretsByFieldValue SOAP API function that caused it to return a 500 error.
- Fixed bug where email filters click through approval links.
- Fixed an issue where OIDC platform connection failed for previously imported users after a domain change.
- Mitigated the possibility of an error in SSH Proxy command processing.
- Removed parameters from ASRA installer to accommodate long secret URLs.
- Resolved an issue with RADIUS challenges in Secret Server Cloud.
- Session recordings which are invalid due to no data are now recorded as an error to prevent failure upon playback.
- Updated discovery SSH scanners to handle messages coming back without the stdout marker.
- Updated logging around Azure AD Sync to make it clearer when the sync stops due to configured groups missing in Azure AD.

Future and Recent Deprecations

Note: This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server: 11.4.000000 EA Release Notes

Release Dates and Notes

On-Premises: February 21, 2023 Cloud: February 11, 2023

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.3.0

Protocol Handler: 6.0.3.26

Known Issues

The distributed engine (DE) package that came with Secret Server (Cloud and On-Premise) 11.3.x prevents Secret Server from performing a DE auto-upgrade—making a manual upgrade necessary. See the Distributed Engine Auto-Upgrade Does Not Work bulletin for details.

New Features

User Interface Streamlining: Classic UI Removed

The classic UI is no longer available as an option, and can no longer be enabled. This followed notifications of phased deprecations in prior releases. Several improvements have been made to the UI based on feedback from customers regarding this change.

Checkout Extension Maximum Limit

We created a global configuration setting that allows administrators to set a maximum secret-checkout extension interval. This provides additional admin control by specifying granular limitations to users extending a checked out secret. The time limitation begins at the point of checkout extension and extension time defaults to the set checkout time

Disaster Recovery Enhancements

"Replicated User Status on Disaster Recovery Source" configuration can now be set to:

- Mirror Source, which is the existing behavior
- Disabled by default

New Synchronization Items:

- Character sets
- AD users and groups

Discovery User Experience Improvements

We updated the discovery user experience to reflect the style and design of the application. The legacy pages are still available; however; the new interface items are ready for use, and we welcome feedback on these items. The legacy pages can be accessed by browsing to the relevant new interface and clicking the "View Legacy Page" button. The improvements are:

- Network view is available as a tab on the main Discovery Sources page.
- Network view displays the same results as the legacy page, in a single filterable grid, as opposed to individual tabs for different types of scanners. The new page has the functionality of the legacy page but in a more-

responsive and updated design.

- There is a new filter menu, allowing extensive filtering options of this data.
- Each item in the network view list links to a details page allowing review of the discovered data, as opposed to being viewed inline.
- The grid data is exportable as a CSV, similar to other grids.
- Scanner, scan template, command set and search filter configuration are available from the Discovery Configuration Options > Scanner Definition button on the Discovery Configuration page.
- Source scanner configuration is available in the Discovery Source Configuration page as a tab.
- We extensively redesigned the scanner configuration UI to make this experience more intuitive, and scanners are now displayed in a workflow view.

Generated and Created Password Improvements

Password Complexity Indicator

There is a new visual indicator in the password complexity rules that provides the user with a better understanding of the strength of their password. The combined score considers both entropy score (brute force defense) and character limitations (social engineering defense). In the case that the score is deemed too low, the UI provides recommendations to the user on how to increase password strength.

New Password Rules

We introduced character rules to password complexity selection to enhance the strength of generated and created passwords, if enabled. The new rules provide flexibility in the granularity of the rules. Each selection impacts both entropy and overall strength score. The rules include minimum characters from:

- Lowercase letters
- Symbols
- Numerals
- Uppercase letters

Opt-In Engine Upgrades

Distributed engine upgrades are no longer mandatory for every release. We added a new setting to the Distributed Engine Configuration page to set the minimum required engine version. Modifying this will trigger an automatic update for any engine below this version.

In the action menu for an engine on the Sites page, a manual upgrade can be triggered for individual engines below the latest version, which prompts the engine to update when it next calls in.

When changes are made needing an upgrade, the minimum required version is updated during the update process, and all engines update immediately.

"Run Scripts" Role Permission

We created a new "Run Scripts" role permissions to separate privileges in script management. Holders of the "View Scripts" role permission cannot execute test runs of scripts, and the new role permission must be assigned to

perform this task.

Administer Scripts remains unchanged and allows view, edit, and run permissions.

Syslog Timestamps

There is a new setting in Syslog configuration allowing the selection of timestamp formatting. The standard for Syslog indicates that ISO timestamps should be used; however, some consumers use the legacy format. There is now a selection between Syslog and ISO format. Syslog will be the default for upgrades to allow current configurations to retain their behavior, and ISO format is the default in new instances.

Site-Specific FIPS Configuration

Individual sites are now configurable for FIPS compatibility. The setting is available on the Administration > Distributed Engine > Site configuration page, in the Engine Default Settings dialog box. All engines on a site will use this setting, overriding the global setting, which is configured at Administration > Configuration > Security.

Enhancements

- Added a configuration option to disable the SMB heartbeat fallback check.
- Added a secret policy setting to control "Change Password Upon Check-In" behavior. Previously, this was automatically enabled if "Require Check Out" was enabled.
- Added additional debugging output for SSH proxy when using the "ALL" logging level.
- Added audits for emailing and downloading reports.
- Added endpoints for Update Password Type Auth, Get Password Type Auth, and Create Password Type Auth. These allow you to create and update records for the command arguments on RPC command set up.
- Added the configuration setting "Allow Files without Extension" to the configuration preview.
- Added the internal site connector configuration to the configuration preview.
- Added the User parameter to the IBM iSeries Mainframe connection for launching, password changing, and heartbeat.
- Bulk edit share now has a "None" permission which allows removing permissions.
- Changed IIS web.config configuration to disallow access to the file uploads folder.
- Enabled more connection classes to use read-only mode.
- Enhanced secret export logging.
- Improved performance of dependency matching within discovery.
- Improved performance of the role assignment page and added a user panel on the same page.
- Improved performance of the secret to computer matching operation that runs as part of discovery.
- In the data replication summary log now lists in alphabetical order, success and version number will appear before the list of items and any errors are appended at the end.
- Optimized application caching.

- Updated the new UI to allow newly generated SSH keys to have a blank passphrase, which matches legacy UI functionality.

Bug Fixes

- Fixed a memory leak in SSH proxy.
- Fixed a SAML audit error.
- Fixed an error that occurred when multiple identical domains were created.
- Fixed an error where the new SSH proxy custom SSH cipher suite settings were not picked up by distributed engines.
- Fixed an issue in the heartbeat status by day report that would cause the same secret to be counted twice on days where the secret transitions between heartbeat failure and success.
- Fixed an issue where "Days Until Expiration" value on the secrets grid would show a large negative number if expiration is forced. This now displays "Expiration forced."
- Fixed an issue where "requires approval type" could not be set by policy.
- Fixed an issue where a user with edit permissions could not rename a folder.
- Fixed an issue where an "invalid SQL error" was incorrectly displayed when a report timed out.
- Fixed an issue where an error was displayed in an edit field dialog box.
- Fixed an issue where an inbox notification was not clearing in Secret Server Cloud.
- Fixed an issue where completed master encryption key rotation would not show as such.
- Fixed an issue where converting a secret in a folder with a launcher settings policy threw an error.
- Fixed an issue where deleted Active Directory groups were not correctly marked as inactive when synchronized.
- Fixed an issue where disaster recovery would log many password requirement errors.
- Fixed an issue where failing Syslog/SIEM messages did not respect updated Syslog Server configuration.
- Fixed an issue where file contents during SFTP/SCP file transfers were included in the session keystroke recordings.
- Fixed an issue where installation on specific dates on servers with a dd/mm/YYYY localization configuration would prevent some configuration settings from being read.
- Fixed an issue where Local site could not be configured to use Custom SSH Cipher Suite settings when set to process on the Web Site.
- Fixed an issue where manual backup did not work in maintenance mode.
- Fixed an issue where master encryption key rotation would fail due to discovery import rules running at the same time.
- Fixed an issue where missing file attachments caused DR replications to fail.
- Fixed an issue where monitoring and termination of live sessions was not displayed in the UI. This now takes the user to the regular session playback page, which displays the live session.
- Fixed an issue where PowerShell-based dependency changers would not correctly pass arguments.

- Fixed an issue where PuTTY would close immediately following a session error. This was due to a default setting change in PuTTY, which is now explicitly set to remain open on installation or update of the protocol handler. This requires a protocol handler update.
- Fixed an issue where reports generate an application error if users navigated away from the report while it was loading.
- Fixed an issue where scrolling the secrets grid view would deselect items.
- Fixed an issue where Secret Server did not correctly react when two templates have the same field if one had spaces that the other did not.
- Fixed an issue where secrets would not open when users have folder view and secret list permissions. The secret audit within that folder should be accessible.
- Fixed an issue where session recording would sometimes show a 500 error, even though the client would retry. Replaced this with a HTTP 429 response, explicitly informing the client to retry.
- Fixed an issue where session recordings could not be saved to a UNC file path due to missing permissions on the root of the path.
- Fixed an issue where sorting by folder path in the secret grid view would return an error.
- Fixed an issue where SSH dependencies would not process on distributed engines.
- Fixed an issue where SSH Proxy would not allow a launcher to connect in maintenance mode. This is now possible in non-recorded sessions—recording is not possible in maintenance mode.
- Fixed an issue where the advanced session recording agent would attempt to make many reconnections in a short time span.
- Fixed an issue where the 64-bit protocol handler would not function when "Enable Protocol Handler Auto-Update" was enabled.
- Fixed an issue where the folder picker would not populate when adding folders in event pipeline policies while in unlimited admin mode.
- Fixed an issue where the IBM iSeries password changer was not properly adding the model value to the connection string.
- Fixed an issue where the light/dark mode toggle displayed "Enable Dark Mode" even though the UI was already in dark mode. This was due to a dark mode browser preference and no explicit user preference having been set.
- Fixed an issue where the SAML AuthnRequest was sending a blank RequestedAuthnContext when Authentication Context was set in the Identity Provider Configuration.
- Fixed an issue where the Secret Server website would not load if the internal site connector is unavailable at startup.
- Fixed an issue where the terminate option on the session playback page was missing.
- Fixed an issue where the test buttons would not function for the Oracle Account Ver. 2 password changer.
- Fixed an issue where the UI session monitoring search required additional permissions to load.
- Fixed an issue where the unlimited administrator watermark could block interaction with some page elements.

- Fixed an issue where the wrong error message would be shown when trying to apply an invalid data source key under data replication.
- Fixed an issue where user permissions on replica instances were removed erroneously when data replication ran.
- Fixed an issue with check in when the "Check In Secret on Launcher Close" and "Close Launcher on Check In Secret" settings were both false.
- Fixed an issue with DR replication where some operations would give an error "The incoming request has too many parameters."
- Fixed an issue with excessive CPU usage for RDPWin.exe. Protocol handler and session connector no longer track or record processes using WMI. Now they use native Windows calls, reducing CPU usage of the Windows WMI Provider. The exception is when "Run as secret credentials" is used—it still uses the WMI process tracking.
- Fixed an issue with replicating domain users. This now correctly links the user with the replicated domain.
- Fixed an issue with slow loading sessions failing to load when using session connector. This required an update to the latest protocol handler (RDS) on the session connector server.
- Fixed an issue with the group filter in event pipelines to ensure precise name matching is correctly used.
- Fixed an issue with the SearchSecretsByFieldValue SOAP API function that caused it to return a 500 error.
- Fixed bug where email filters click through approval links.
- Fixed an issue where OIDC platform connection failed for previously imported users after a domain change.
- Mitigated the possibility of an error in SSH Proxy command processing.
- Removed parameters from ASRA installer to accommodate long secret URLs.
- Resolved an issue with RADIUS challenges in Secret Server Cloud.
- Session recordings which are invalid due to no data are now recorded as an error to prevent failure upon playback.
- Updated discovery SSH scanners to handle messages coming back without the stdout marker.
- Updated logging around Azure AD Sync to make it clearer when the sync stops due to configured groups missing in Azure AD.

Future and Recent Deprecations

This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server Cloud Release Notes

These are the release notes for recent Secret Server Cloud versions.

For changes pushed to cloud between major releases, see the Secret Server Change Log.

Secret Server Cloud Release Notes for April 5, 2025

Cloud Release Date: All Regions: April 5, 2025

This release was previously dated March 29, 2025.

For changes pushed to cloud between major releases, see the Secret Server Change Log.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.43.0

Protocol Handler: 6.0.3.33

If your protocol handler version is 6.0.3.26 or lower, you must manually upgrade to a higher version. Automatic upgrades will not work for versions 6.0.3.26 or below. However, if your protocol handler version is 6.0.3.27 or higher, the automatic upgrade will function properly.

New Features

Azure Key Vault Integration

Azure Key Vault Integration (AKVI) simplifies management and governance of NHI's and secrets from the CSP's native vaults. With AKVI you can centrally manage and update secrets to one or more Azure Key Vaults and rotate passwords or values more frequently. With fine grained roles and permissions, audit and logging, AKVI provides increased governance, visibility, and awareness of secrets managed in Azure Key Vault without affecting development velocity or processes. AKVI is available on Secret Server Cloud, the Delinea Platform, and Secret Server On Premises.

Additional Approval Workflow Type

A new approval workflow type is available, allowing owners to bypass approval while ensuring approvers still require it. The "Standard Including Editors and Approvers (Owners do not need approval)" option offers more flexibility in approval processes to meet organizational needs.

Bulk RPC on Secrets with Checkout Enabled

Bulk RPC actions are available to secrets with checkout enabled. This feature uses random passwords during bulk operations, ensuring that passwords remain secure and hidden, even during bulk updates, without compromising secret integrity.

Bulk Update Secret Fields

Bulk updates for secret fields are now available, enabling users to edit and update multiple fields across secrets in the folder view. This simplifies importing and formatting secrets, streamlining secret management for large datasets.

Change Password on Checkin Configurable Limitation

"Change Password on Check In" is now more flexible. Users can now configure the system to retry password changes a specific number of times before allowing the secret to be checked in without a change, streamlining access when password changes fail.

Global Manual Approver Workflow for Ticketing Systems

A manual approval workflow is now available for scenarios where the primary ticketing system, like ServiceNow, is unavailable. This fallback option ensures that users can still gain access to secrets through a manual approval process, maintaining workflow continuity even during system outages.

PowerShell 7 Support for Scripts

Secret Server now supports PowerShell 7 scripts, allowing users to run both legacy PowerShell scripts and PowerShell 7 scripts. This update ensures compatibility with the latest thycotic.secretserver module and helps avoid disruptions from version conflicts.

PowerShell Ticket Integration—User Information Passed as Arguments

PowerShell ticket integration has been enhanced to pass user information (userID, username, and email) as arguments in scripts. This update provides greater flexibility for ticket validation, enabling more customized and user-specific logic in ticket-related actions.

Pre-Compiled Version of Secret Server On-Premises

A pre-compiled version of Secret Server for on-premises deployments is now available. This version allows files to be signed through catalog signing, addressing code integrity violations and ensuring compliance by maintaining integrity and trust standards for all files.

Secret Icons

Secret Icons allows you to display icons for secrets in the secret list, and secret details page. Icons can be set at both the secret and secret template levels.

Fixed Issues

405016	Fixed: The RPC by Day report is now formatted to user's time zone.
530294	Fixed: Key rotation failure. We now allow a particular password type to be set for account take-over when importing an account into Secret Server.
537916	Fixed: The folder-permission API now requires view or administer folder permissions to query by a folder ID. Previously, you could also do this with the personal folders role permission.
539187	Fixed: Secret access request viewing very slow. Bug fixed with loading large numbers of secret access requests.
546108	Fixed: Add a "Matches" tab on discovery account rules to show computer accounts that match the defined rule condition. This replaces the account rule filter on the network view that has been removed.

556742	Fixed: An issue that prevented empty dependency groups from being deleted.
557774	Fixed: Errant heartbeat every five minutes. When creating a new secret template with heartbeat enabled, if you do not change the interval value it will now correctly assign heartbeat interval of 60 minutes.
559102	Fixed: An issue where large-item-count folder searching was broken.
560138	Fixed: The secret template fields grid would not always load all records properly if there were more than 60 fields on a template.
564689	Fixed: User appearing locked out after the lockout period. On the User General page (admin page), added an Unlock User button and chips and messages to reflect lockout interval for user locked out by failed logins.
566423	Fixed: Resolved an issue where downloaded report names appeared garbled when the language was set to Japanese or Simplified Chinese. Fixed: An issue in the Platform where downloaded reports were incorrectly named "null." Reports now display the correct filenames based on the selected language.
571212	Fixed: Test Script page not working. SQL parameters now work in the new UI.
571231	Fixed: Role audit log error. The Action field of role audit logs now display correctly when the log is created in a language that uses Unicode characters
575503	Fixed: A dependent library used in SAML in Secret Serverhas been updated to close potential security vulnerabilities. It uses a different version of a saml.config when using the legacy SAML configuration, and a conversion process to update saml.config has been added to the upgrade system. Please see Secret Serverdocumentation "Troubleshooting SAML Configuration Errors After Upgrading" if using a saml.config file and having issues.
578291	Fixed: Session recording search times out.
578890	Fixed: Removing a launcher having multiple secrets linked will no longer fail.

580299	Fixed: An issue with /api/v1/launchers/secret endpoint throwing errors with complex URLs.
581180	Fixed: An error when checkout had expired, switching to the settings tab on a secret would throw a red banner error instead of redirecting the user to the checkout page.
582171	Fixed: Double email notifications for access approval request. Access request emails will now indicate a status of the workflow. Viewing a workflow online will also render a visualization of the workflow status.
582538	Fixed: An issue where session messages sent from Secret Serverwould not show during RDP Proxy sessions.
582728	Fixed: An issue where users who had permission to edit secrets could not toggle auto-change using the bulk action on secrets that should have allowed it, based on the permission set in the secret's template.
583939	Fixed: Incorrect active session display. Secret active launcher sessions now updates list when a session is launched from the page.
585609	Fixed: error when removing scanners from a discovery source. Added a custom exception for "scanner X already added" and UI refresh to stay in synch with back end.
591272	Fixed: An issue with the Launcher template when modifying the field "Use SSH Tunneling with SSH Proxy."
593801	Fixed: An issue with SSH key integration expiration configuration.
595169	Fixed: An issue where there was no option to add a step in a workflow. It is no longer possible to delete the last step from a workflow. Existing workflows with no steps will now display a default starting step when opened to edit.
595565	Fixed: An issue where dependency changers in SSC were not passing arguments to scripts, resulting in empty output files. Dependency changers now correctly pass arguments, and the status no longer incorrectly shows as disabled.
599173	Fixed: In active sessions inside a launched secret, when the username that launched the secret contains Unicode characters, they displayed incorrectly.
601706	Fixed: Intermittent Azure message loss. Implemented retry logic for publishing to Azure service bus queues.

603681	Fixed: Resolved a password display Issue (with "comment required" enabled) where, after waiting on the Overview tab for 5+ minutes, the password was displayed as [object Object] instead of prompting for a comment again. Users are now correctly required to re-enter a comment when accessing the password.
603779	Fixed: An issue with category and report permissions showing 0 items when permissions are assigned to users.
605053	Fixed: Heartbeat and password reset failures. Added more support for expired AD account password changers. Secrets that use an AD privileged password changer to rotate the password for an expired AD account will successfully complete the rotation process. Previous behavior involved rotating successfully and then failing the verify step, resulting in the new password not being saved on the Secret Serverside. Subsequent heartbeats may fail for the secret since the account is expired. Password Changes using a secrets own credentials may fail as well.
607434	Fixed: Discovery analysis SQL timeouts. The query that populates the discovered account metrics has been made more efficient. It should no longer have timeout issues.
608395	Fixed: Columns that should have been hidden were selectable in the column selector. If selected, they would (incorrectly) display until page reload. The conditionally available columns are now correctly set visible or hidden in the column selector.
614002	Fixed: Turkish not displaying correctly in email. Turkish characters should publish correctly in email HTMLs.
614465	Fixed: Tooltip location on the Launcher Configuration page.
616185	Fixed: Resolved an issue where users could add members to a migrated group in SSC via individual user modifications. Now, all membership changes must be managed in the Platform, ensuring proper access control.
616221	Fixed: In SSC the from email address field in email configuration settings is restricted to the secretservercloud domain and the TLD excludes .co.uk as a valid option. On premises instances will only validate that it is a valid email address format but will allow any domain to be input.

617344	Fixed: An issue with password that contains username and added a new item to the local-user password configuration area that optionally prevents the password containing the username.
617429	Fixed: An issue where an invalid version number could cause Secret Server to become unresponsive.
617445	Fixed: Updated command sets to no longer add extra spacing between lines, and added validation around comments in command sets, instead of auto-removing extra comments, to reduce confusion on save.
617607	Fixed: Discovery services grid did not sort. Added computer Services API endpoint so the computer services component now has paging and sorting.
618528	Fixed: Resolved an issue where secret policy settings were not properly inherited by secrets, causing discrepancies in the Approval page. Additionally, the "Language Resource Not Found: OnlyOptionViaPolicy" message has been fixed. Secret policy settings now correctly apply as expected.
618869	Fixed: An error with "Default Only" on RPC schedule on a active secret policy.
619554	Fixed: On-Premises Secret Server instances with PRA will now get emails to the Secret Server instead of the Platform instance.
620165	Fixed: Display to show <tenantname>.delinea.app when opting into a prod instance.
620338	Fixed: An issue with "minimum required character count" rules options containing an invalid choice.
621226	Fixed: Resolved an issue where repeated execution of Entra ID secret heartbeats would cause a "Headers too long" error.
621935	Fixed: Resolved an issue where viewing a secret with MFA enabled incorrectly logged "Password Displayed" in audit entries. Now, the audit log correctly records the action as "View" when no other interactions occur.
622254	Fixed: Resolved a bug where when Platform is integrated with Secret ServerCloud and Open ID Connect Platform login is used–in some situations the redirected Platform login page would be incorrect.

622479	Fixed: Error during opt-in in PIC for Europe region.
626465	Fixed: Audit with no notes. Secret policies should no longer create an empty audit log when modifying launcher settings but reverting changes before saving.
626702	Fixed: An issue with users not being re-enabled when logging in through Platform after being disabled by automatic user management.
627109	Fixed: Launchers filter was incorrectly labeled was "Template."
627246	Fixed: The field header example on secret import now wraps correctly.
627291	Fixed: The "All launchers" option of the launcher filter now returns all results as expected.
627619	Fixed: Changed database cleanup logic which was causing some heartbeat/RPC audit records for inactive secrets to be removed before the "Max Secret Log Length" was reached.
627731	Fixed: If an Entra discovery source is created in Secret Server and Platform integration is configured with Inventory Forward enabled, there was a bug when deleting roles from the Discovery Network View in Secret Server. It would cause Entra roles to show up in Platform inventory.
628439	Fixed: Corrected a typo on Launcher Mapping page.
629517	Fixed: Resolved an issue from the previous update where toggling an Active Directory account's expiration status could prevent verification after a password change.
629584	Fixed: An issue with the password compliance check notification on a secret.
630728	Fixed: A bug where saving an approval method for a secret does not persist correctly.
631133	FIxed: An issue with AD privilege password changing partially failing for accounts that had null values for "accountExpires."
634286	Fixed: Issues with dropdown options causing enum values to be displayed for the Secret Security Approval Type so that correct localized strings are displayed.

634484	Fixed: Excessive CPU usage by correcting the SessionKey parameter to varchar, eliminating implicit conversion.
635135	Fixed: The "Ignore permission errors" checkbox is now available without a page refresh.
637136	Fixed: Do not include azure domains or inactive AD domains in group-type precheck.
637139	Fixed: Users missing their ExtendedUserMapping will register as a warning in the precheck instead of an error.
637373	Fixed: Corrected regression where secrets were counted twice in some scenarios. Secrets grid count functionality restored. Count ("# Items") is the sum of the number of first-child subfolders and the number of secrets in the current folder.
637690	Fixed: Incorrect GUI label. Updated Log Level filter label from "Site" to "Log Level."
638073	Fixed: A bug that prevented the secure-platform-access step in the PIC from auto-skipping.

Improvements

546156	Improved: Logging in when MFA setup is required now immediately redirects the user to configure MFA.
566484	Improved: Reports page size minimum has been increased to 60.
575905	Improved: Updated scanner template creation UI to combine both OU Input templates and non-OU Input templates into the same dropdown in categories.
580253	Improved: No dynamic update for active sessions. Active launchers section on secrets now updates every 30 seconds to show an updated list of active launchers.
582378	Improved: Updated grids using timestamps to use datetime in order to properly respect user preferences.
586608	Improved: Password validation failures for common number substitutions. Dictionary now indicates "Dictionary words including common number substitutions."

593543	Improved: Removed time zone tooltips from reports to reduce confusion when time zones are set by the report.
594323	Improved: Empty pinned folders now inform users of the empty sections.
603721	Improved: Resilient secrets log text information and added operation progress percentage estimation.Fixed: An issue where resilient secrets operation was not interrupting on operation timeout.
605210	Improved: Performance of discovery scanner delete: increased timeout to 24 hours, lowered isolation level where possible, and added logging for each delete operation.
609101	Improved: Added a non-configurable 30-second secret password timeout to improve security and reduce stale password issues. The timeout applies to: ■ Show password—hides after 30 seconds (previously visible forever). ■ Copy password—password value cached for 30 seconds only. Copy password icon clicks after that will trigger a fresh API call (to reduce stale password conflicts). Note that this does not affect the value that was already copied to the clipboard, only what will be copied when the user clicks the copy password icon.
610739	Improved: Performance improvements were made to the "Shared with me" Secret, and "Browse All Folders" views for customers with a large amount of folders in a highly nested structure.
611190	Improved: In the secret template list field, list and URL list now have their "dispose for display" boxes checked by default to denote the data's plaintext status. The expose for display control is also disabled so that it can not be unchecked by accident.
611573	Improved: Updates to Discovery Scan Status Report Query.
612177	Improved: Error handling for Platform credentials that become invalid.
612739	Improved: A new setting, "Disable Legacy Bookmark Pages," has been added the admin/user experience section. This setting is false by default. When true, the legacy bookmark pages used by legacy WPF will be disabled. This allows administrators to disable the setting and ensure they do not have any of these legacy clients that require it. This setting will default to disabled in a future release.

614508	Improved: Added the ability to view the Active Directory group type. This is displayed under the General tab of a group. If a group does not have a type, it will display as a hyphen. Otherwise, it will show one of the following: Global, Universal, or DomainLocal.
616620	Improved: Secret export now audits the current "Export" action and a new "Export retrieved" action to indicate that the user actually retrieved the file. Previously, you could close the browser window before retrieving the export file.
618004	Improved: Through the user experience settings, a user can use a new "COMMENT" audit action to separate the VIEW action of a checkout (or ITMS) protected secret from the required comment.
619512	Improved: Event pipeline set-custom-value tasks can now increment or decrement pipeline variables by custom values.
619515	Improved: Added additional group type validation (empty group types and DomainLocal group types) for the Data Sync step in the PIC. If empty group types are detected, users are prompted that there are empty group types and are instructed to run the directory services sync. For any DomainLocal groups found, an error message appears in the pre-check table within the Data Sync step stating that DomainLocal groups are not supported.
626041	Improved: Modifications to build pipeline to precompile, copy and overwrite Secret Serverprecompiled assets to webroot folder. Packaging and installer remain as they were prior to PR.
626668	Improved: With Platform Integration, support for the setting "Create Groups during synchronization" is completely deprecated. Now, all Platform native groups will be created automatically and any directory groups through Platform need manual linking.
626881	Improved: Updated discovery import rules to prevent duplicate account creation and unintended unlinking of service and Active Directory accounts. Added broader test coverage, including integration tests, to ensure correct matching, unmatching, and unchanged behavior when re-running imports.
627169	Improved: Updated discovery analysis layout to emulate dashboard styles and to better accommodate large datasets.
627297	Improved: Added a new date and time range filter to Session Monitoring.

627303	Improved: Added a new state flag to indicate whether the Delinea enablement code has been entered.
627304	Improved: Added a new state flag to indicate the customer has completed the Platform integration.
627344	Improved: The inbox template editor now includes options to select message properties by selecting which message. This helps clarify that only message properties on the targeted message are available to merge into the template.
631776	Improved: The application picker that appears when both Secret Serverand Privilege Manager are installed has an updated design. This slightly increases the speed of the login process.
633054	Improved: Optimized memory management to reduce latency buildup in the US BGW, preventing performance degradation over time. Restored ASP.NET metrics for better visibility into garbage collection and CPU usage.
635732	Improved: Add PasswordTypeIds as a filter on the api/v1/secret-templates-list endpoint.
635803	Improved: Only the users that are migrated are validated.
636130	Improved: When enabled, "Show secret icons" in User Experience will display icons for secrets in grid, card, and detail view.
638928	Improved: Removed legacy secret access request aspx pages.

Secret Server Cloud Release Notes for August 3, 2024

Release Date and Notes

Cloud Release - All Regions August 3, 2024

For changes pushed to cloud between major releases, see the Secret Server Change Log.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.33.0

Protocol Handler: 6.0.3.28

With this version, protocol handler has received changes to core internal functionality that prevents automatically updating to version 6.0.3.27 from a prior version. In environments with protocol handler automatic update enabled, the protocol handler will automatically update to version 6.0.3.26. To use the latest functionality and fixes of protocol handler, you must redeploy or install version 6.0.3.27 to end-user machines. Following that update, the automatic update will continue to work as before.

New Features

Entra ID Discovery

We are excited to introduce Entra ID discovery in Delinea's Secret Server! This enhancement expands our current discovery feature by adding support for Microsoft's Entra ID, alongside our existing AWS and GCP discovery types.

With Entra ID discovery, Secret Server can now scan Microsoft Entra ID for roles and users, importing users as secrets based on the Entra ID User Account template. This completes the suite of features necessary for Secret Server to discover and manage accounts from Microsoft's Entra ID.

Bug Fixes, Changes, and Enhancements

Bug Fixes

Fixed: Discovery runtime summary information is now correctly accessible for screen readers.

Fixed: Mobile logo now displaying properly.

Fixed: The SSH key-expiration configuration value now displays correctly.

Fixed: Updated the distributed engine service to persist the current the web-proxy.config file upon update. When upgrading to version from version 8.4.29.0 or lower, the web-proxy.config will be overwritten, but any upgrades afterwards will preserve it.

Fixed “Secret Erase” translation in some non-English languages.

Fixed a bug where a distributed engine ignored the “tasks Should Respect MaxShells restrictions” setting.

Fixed a bug where distributed engines ignored MaxShells restrictions.

Fixed a bug where distributed engines ignored WinRM quota limits.

Fixed a bug with running disaster recovery data replication from an older source to a newer replica.

Fixed a bug with the export/import settings where it was not resetting after leaving the page.

Fixed a check-in bug that caused a red banner warning.

Fixed an issue during Platform group synchronization where groups with long names would cause an error.

Fixed an issue so users and roles now always show in SSC, even if in unified mode, but still are hidden when using platform.

Fixed an issue to restore configurability of secrets associated with custom launchers.

Fixed an issue where deleting computers from the discovery network view failed to show a confirmation dialog box before continuing.

Fixed an issue where enabling QuantumLock on a secret threw the error “The partner transaction manager has disabled its support for remote/network transactions.”

Fixed an issue where stored data growth impacted proxy sessions. The secret session table is now managed and part of the supported tables of the data retention feature. Secret session records are now truncated in accordance with the existing data retention configuration. Please make sure to review your organization's Data Retention “Max Record Age” settings.

Fixed an issue where the “All time” filter on the inbox might not show all results.

Fixed an issue where users with MFA enabled would be incorrectly sent to the home page on login, instead of the page they were attempting to access.

Fixed content security policy fields for frame-ancestors.

Fixed incorrect access checks concerning reports.

Fixed incorrect Secret search totals when filtering by multiple templates.

Fixed issue where the “su -id” command was failing when the user did not have access to view the password for the secret they were elevating to.

Fixed issue where the maximum log Length was not used to truncate the tbSystemLog.

Fixed issue with “What folder permissions exist” report. Groups with no active users now properly included on the report

Fixed main navigation alignment issues.

Fixed ServiceNow allowed status validation over distributed engine.

Fixed the “view detail” link on the user detail panel.

Fixed The folder tree is now updated when unlimited admin mode is toggled.

Fixed timeouts for large amounts of data—paging for user audits is now done in the database.

Fixed: A user that did not have the “view launcher password” role permission was unable to create a secret that had a required password because the password field was hidden.

Fixed: Added null checks for username.

Fixed: Added support for Cisco devices when using a question mark after the command or partial command. This allows Cisco to work as normal, while not allowing the blocked commands.

Fixed: Addressed an issue where a launcher type field that was replicated via resilient secrets would not function with all prompt-able field names.

Fixed: Addressed one scenario where a backend process that publishes session information would error.

Fixed: Adjusted secret overview tab to not use a banner for heartbeat failed.

Fixed: Adjusted Secret tab pending password change status to be a chip instead of a banner.

Fixed: Audit handler was missing the “View Configuration Unlimited Admin” permission as an option.

Fixed: Authentication errors are now 401s for API requests and in Platform.

Fixed: Customers who had Easy Move to Platform had duplicate groups created in Secret Server and the existing permissions from the original Secret Server group were not honored. It now disables this new duplicate group and connects the original group to the Platform group as originally expected.

Fixed: During forwarding of inventory data from discovery in Secret Server to Platform inventory, with large amounts of computers, the processes could time out. Made the database calls more efficient and the process no longer times out.

Fixed: Extended the Migration Center to migrate all active roles.

Fixed: Folder path now shows when specified in secret import preview.

Fixed: Heartbeat listed as “pending” when the heartbeat is actually disabled. This occurred when the pending status did not resolve before the secret was disabled.

Fixed: Improved compatibility with Windows high contrast mode.

Fixed: Improved “Regenerate Platform Credentials” to attempt to forward credentials to connected Secret Server Cloud automatically (behind feature flag).

Fixed: In some scenarios only the first 30 subfolders were loaded on initial load for a single folder.

Fixed: In some scenarios the folder tree would not auto-expand when linking directly to a folder.

Fixed: Left navigation expand/collapse toggle incorrectly labeled for screen readers.

Fixed: Login SSH key menu showing properly in cloud when configured.

Fixed: Newer versions of Safari can now play session recordings in Platform.

Fixed: Pinned folders now re-root the tree to the selected pinned folder.

Fixed: Reduced situations where a check-in error could occur when already checked-in.

Fixed: Removed links to legacy create discovery wizard pages.

Fixed: Resolved secret permission issue when many user and groups had been selected and only the 60 were saved when edited again. Resolved for teams selection as well.

Fixed: Searching in all secrets now shows the full folder path for folder search results.

Fixed: Secret password compliance is now calculated when a password is updated to empty and the password is not required. Prior to this, the secret would maintain the compliance flag that was calculated when the password had a value. A password with some characters might fail compliance, but if there is no password and it is not required, then it is compliant.

Fixed: Site name now wraps instead of truncating on the “sites and engines” page so you can read the whole site name.

Fixed: SQL report editor is now properly announced for accessibility.

Fixed: SSH keep-alives sent to the proxy are now relayed to the endpoint server.

Fixed: Teams group membership removed when more than 60 items in Team.

Fixed: Thycotic One Login Link.

Fixed: Unlimited admin mode audit dialog box is now correctly aligned.

Fixed: Updated all the logs to be warnings and information and to state whether they retried or not.

Fixed: Updated Discovery Network view to better handle extremely large record numbers.

Fixed: User username link was sometimes unusable. It is no longer a link. View details link is in menu and preview panel.

Fixed: When viewing folder targets for event pipeline policies the full path is now shown.

Fixed: Remaining KB links now point to `docs.delinea.com` instead of `delinea.center`.

Fixed: About page links not working.

Fixed: Resolved an issue where approvals that cross a day threshold from UTC could not be requested.

Fixed: Resolved a UI issue with discovery import.

Fixed: Resolved an issue that caused SAML logins to fail, resulting in a rollback of the previous update.

Changes

Change: Admin breadcrumb renamed to Settings.

Change: Corrected license expiration banner link.

Change: Platform now specifies “Secret Server” configuration.

Change: Removed the color mode toggle from the top navigation as it is available under user preferences.

Change: The delinea.vault/secretserver/access permission has been removed. This no longer controls Secret Server access for Platform users.

Change: The SSL menu item is removed as it is not an option that can be modified in cloud.

Change: RequirePlatformMfa field is now deprecated.

Enhancements

Enhancement: Added “RPC PRIVILEGED SECRET UPDATED” and “RPC PRIVILEGED SECRET REMOVED” events to audits.

Enhancement: Added “User Lockout Protection” setting to domain.

Enhancement: Added a “Clear cached AD credentials” button in cloud.

Enhancement: Added a “test syslog” button to syslog pages in configuration.

Enhancement: Added a direct link for launching connection manager.

Enhancement: Added AIX support for SSH Proxy su automatic password entry.

Enhancement: Added an OOB RPC template for Okta. Okta requires an “Generic API” secret as the RPC privileged account.

Enhancement: Added an OOB RPC template for ServiceNow. ServiceNow requires an account to have Admin or write permissions to the password field, or an account with those permissions as its RPC privileged account to change the password.

Enhancement: Added DSV links to the Platform settings page.

Enhancement: Added landing page for when the user is unable to access Secret Server instead of showing banners.

Enhancement: Associated secrets will now show “No Access” in the secret name if you do not have access to it.

Enhancement: Converted key management to the latest design and added a verification checkbox confirmation step.

Enhancement: Heartbeat and password-compliance notices now use chips instead of banners.

Enhancement: Improved startup logging for distributed engines.

Enhancement: New import secret page allows you to import when global setting requires that secrets are in folders.

Enhancement: On premise now shows a diagnostics section under settings in the left navigation panel.

Enhancement: The left navigation folder tree now expands on focus to show longer folder names.

Enhancement: Updated password compliance label to a chip.

Enhancement: Updated Putty to version 0.81. Updated version addresses several Putty vulnerabilities, including the Terrapin vulnerability.

Enhancement: Updated Redis library for improved Redis operations.

Enhancement: Updated the server nodes page.

Enhancement: Updated the user profile menu to have more consistent styling and include links to the account details page.

Enhancement: Updated user experience for adding custom logos to Platform instances.

Enhancement: Updated user sorting to cover 2FA.

Enhancement: When a Secret Server is integrated with a Platform tenant, any Platform cloud groups are now automatically and quickly be created in Secret Server to be available for permission delegation.

Enhancement: Aria label added to inline secret-preview copy buttons. Main search category toggles now keyboard accessible.

Secret Server Cloud Release Notes for April 20, 2024

Release Date and Notes

Cloud Release - All Regions April 20, 2024

For changes pushed to cloud between major releases, see the Secret Server Change Log.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.24.0

Protocol Handler: 6.0.3.27

With this version, protocol handler has received changes to core internal functionality that prevents automatically updating to version 6.0.3.27 from a prior version. In environments with protocol handler automatic update enabled, the protocol handler will automatically update to version 6.0.3.26. To use the latest functionality and fixes of protocol handler, you must redeploy or install version 6.0.3.27 to end-user machines. Following that update, the automatic update will continue to work as before.

New Features

Left Navigation Panel

With this release, we have made several improvements to the left navigation sub-menu to provide a better user experience. Some of the most common configuration settings have been moved to the top of the menu. For more information about the latest changes, please see "Main Navigation Drawers" on page 162.

Improved Search

The main search is greatly improved. It now includes content search results as well as users. A status bar shows the full folder path for long secret-folder paths. Highlighted search categories include:

- Content
- Favorites
- Folders
- Secrets
- Users

Entra ID Secret Template for RPC

Secret Server has supported Azure AD remote password changing for several years, this overhaul creates a new password changer and template, Entra ID, that uses Oauth application credentials as a privileged account to change a user password. Entra ID is Microsoft's comprehensive cloud-based identity and access management solution that helps organizations securely manage identities and access across their Microsoft services and applications. Our password changer and template support MFA and conditional-access policies and does not require PowerShell.

Bug Fixes and Enhancements

- Fixed an issue with adding discovery sources that matched the domain of a current secret and were unmatched in the domain-name index table.
- Enhancement: Increased back end performance of event queue processing when there are a large number of inbox rules.
- Fixed an issue where users other than owners could view TOTP backup codes.
- Fixed an issue where OAuth parameters were not validated. The OpenIdConnect flow has been adjusted to validate the redirection URI.
- Fixed issues that could cause incorrect group or user interactions between Secret Server and Platform. We corrected an issue with Platform group synchronization that would not correctly add all group memberships once the number of synchronized groups was over 1000.
- Fixed an issue where a ticket number was not present in SIEM logging.
- Fixed a policy validation issue that occurred when using a $itemvariable.variablename in schedule pipeline minutes.

Secret Server Cloud Release Notes for February 10, 2024

Release Date and Notes

Cloud Release - All Regions February 10, 2024

For changes pushed to cloud between major releases, see the Secret Server Change Log.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.21.0

Protocol Handler: 6.0.3.27

With this version, protocol handler has received changes to core internal functionality that prevents automatically updating to version 6.0.3.27 from a prior version. In environments with protocol handler automatic update enabled, the protocol handler will automatically update to version 6.0.3.26. To use the latest functionality and fixes of protocol handler, you must redeploy or install version 6.0.3.27 to end-user machines. Following that update, the automatic update will continue to work as before.

New Features

QuantumLock

Secret Server's *QuantumLock* is a feature that provides an additional security layer by protecting secret data using asymmetric encryption (a public/private key pair) where the private key is a human-generated password. This feature is independent of regular permissions, Secret Server login access, or physical access to the machine running Secret Server.

A shortcut way of thinking about QuantumLocks is as an extra password for secrets that is held by a set group of users. In addition, both the password and the group of users are reusable for other secrets. In addition, QuatumLocks future-proof our digital security infrastructure against the advancing capabilities of quantum computing.

QuantumLock is an upgrade of the earlier doublelock feature. Besides the name change, the difference is QuantumLock offers the option to use a quantum-safe algorithm for encapsulation to protect the private key, specifically CRYSTALS Kyber-1024, which is designed to counter the potential threat from quantum computers to current encryption methods.

Enhancements

- Enhancement: Added a link to configuration audits on the Remote Password Changing page.
- Enhancement: Added a running log to disaster recovery so progress and duration per table can be tracked during replication.
- Enhancement: Added an event subscription called "Disaster Recovery Replication Success."
- Enhancement: Added auditing of password change schedules.
- Enhancement: Folders in favorites quick access are now filtered when searching.

- Enhancement: Improved HSM cryptography by adding support for AES 256 encryption. This ensures that all keys protecting the secret key will be at the same strength for organizations requiring this level of encryption.
- Enhancement: If an Azure Active Directory configuration in directory services becomes corrupt, you can now view and update the credentials to fix it.
- Enhancement: improved internal security checking around launchers.
- Enhancement: Improved SSH proxy block-command handling in VIM.
- Enhancement: Launching a secret now opens in a dialog allowing launch to occur without leaving the grid or current page. Restricted actions like checkout can be performed in the dialog.
- Enhancement: On the Proxying Configuration page, you can now automatically generate new SSH proxy host keys.
- Enhancement: Platform configuration settings were added to disaster recovery.
- Enhancement: Secret search performance improvements. The secret grid now only requests extended fields that are showing. When column selections are updated, a new request is made only if the extended field choices have changes.
- Enhancement: Secrets grid modal on the Secret Erase Requests search page now auto-scrolls.
- Enhancement: The login policy now supports line breaks.
- Enhancement: The secret search API now has a comma-delimited filter parameter for template IDs, which allows searching beyond IIS URL limits compared to the existing array version. Both are still available.
- Enhancement: The user profile allows for date and time format setting.
- Enhancement: Updated the toast message displayed when saving user preferences to accommodate screen readers.
- Enhancement: Users are no longer redirected from the licensing page.
- Enhancement: When Secret Server Cloud is Platform integrated, there is now an "Add from External Directory" option in secret sharing that allows searching directory sources from Platform to add users or groups.
- Enhancement: Added new optional parameter "nobus=true" to the healthcheck endpoint. This allows a faster response in situations where no lookup of the bus status is required.
- Enhancement: Adjusted the password compliance validation job to process more secrets on each run.
- Enhancement: Discovery port scanner now aborts if elapsed time expires prior to windows TCP handshake. Discovery port scanner will now also log a helpful message if the windows TCP stack aborts due to reaching the windows internal max syn retry count.
- Enhancement: The schedule pipeline task in event pipeline policies now supports using a variable for the schedule delay input.
- Enhancement: Unlimited Admin can now check in checked out secrets by other users.
- Enhancement: Added a Computer Scan Results tab to discovery.
- Enhancement: Added a new option to the Distributed Engine page for configuring "pending engines" that allows a pending engine to be assigned to a site without activation.
- Enhancement: Added a note to audits when the system disables a Secret Server user.

- Enhancement: Added an SDK link.
- Enhancement: Added the table tbTerminalConnectionHistory to the list of tables that is handled by the database cleanup consumer. It will periodically delete any records over a certain age, which can be customized by the user.

Bug Fixes

- Added "view all folders" link that appears when folders are filtered in a pin view.
- Added a download button for session recording to Secret Server. The change does not appear for vault sessions in Platform.
- Added aria labels to the notification bell to support screen readers.
- Added new REST API patch method to controller which calls pre-existing latestversion.txt processing code.
- Added protocol handler step-up upgrade. Protocol handler will not try to upgrade versions 6.0.3.26 to newer versions as they must be updated manually. Released new 6.0.3.27 version which will be able to upgrade to future versions.
- Adjusted license tracking for session-recording-enabled secrets so that secrets that have no launchers are excluded.
- Adjusted organization of some administrative menu items in the configuration preview.
- Adjusted permissions on Session Monitoring page so that users with "View Own Session Recordings" permission will only see their own recordings.
- Adjusted the display of administrative items from Platform to avoid perceived duplication.
- Adjusted the log level downward for certain engine messages for syslog to avoid overloading the engine log table.
- Applied a more reasonable default SQL timeout.
- Clarified explanatory information on the Secret Import page to highlight that file fields are ignored.
- Converted dependency template management section to new UI.
- Converted Initial User page to the new UI.
- Corrected an issue where the Distributed Engine page did not respect the "Deleted" filter.
- Disabled the legacy bookmarklet pages.
- Disaster recovery now migrates teams.
- Fixed a client-side error on the Secret Settings page when viewed from Platform.
- Fixed a display issue on the IP Address restrictions page.
- Fixed a missing localization-key issue.
- Fixed a visual bug on secret templates so the password type dropdown no longer appears as "None" if a password type has been set.
- Fixed an edge case that could result in duplicate disabled usernames, possibly causing DR conflicts.
- Fixed an error that could occur on the Advanced Session Recording page.

- Fixed an HTML-encoded document link in discovery scanner.
- Fixed an issue an erroneous warning popup appeared saying a distributed engine is required for Active Directory when the SSC cloud instance has "Azure AD Domain" as the only domain.
- Fixed an issue on the Admin Roles page where the edit button for role permissions was mistakenly requiring "Administer Role Assignment" instead of "Administer Role Permission."
- Fixed an issue that could cause an incorrect error message to display when using the SQL report editor.
- Fixed an issue that could cause the secret picker to display with a horizontal scroll bar.
- Fixed an issue when searching in Secret Share with the "Add from External Directory" option with results of more than 2100 groups would throw an error.
- Fixed an issue where a proper validation message may not display when trying to give a duplicate name to a group.
- Fixed an issue where a secret erase request could no longer be canceled.
- Fixed an issue where banner text referenced only "engine," which was potentially confusing. It now mentions "distributed engine" explicitly.
- Fixed an issue where created hooks would not display on the secret.
- Fixed an issue where enabling RPC on a template through the API could impair the template's functionality.
- Fixed an issue where existing linked groups under the Platform Integration area on the Groups tab would not load.
- Fixed an issue where if a non-local site was used to send syslog to the syslog server any failure was queued back into the database (tbsyslogfailedmessage) and resent indefinitely. This has been resolved. Additionally, we implemented a syslog circuit-breaker system if a non-local site is used to prevent flooding the message queues with syslog messages when failure is expected.
- Fixed an issue where localization load requests would await indefinitely in some cases.
- Fixed an issue where pinned folders would not be removed when the corresponding folder was deleted.
- Fixed an issue where Platform synchronization was running too frequently in some cases.
- Fixed an issue where renaming or copying the "Oracle Account (Template Ver 2)" secret template caused password changes to fail.
- Fixed an issue where Resilient Secrets (DR) sent secret field launchers across the wire for every replication.
- Fixed an issue where selecting Generate New SSH Key on a secret would not generate a new SSH key.
- Fixed an issue where sorting the launchers list by name could display duplicates.
- Fixed an issue where the checkout screen could briefly show while a secret is loading.
- Fixed an issue where the child launcher type was not always visible on the new custom launcher page.
- Fixed an issue where the Everybody group from Platform would not match up properly with the Everybody group from Platform User sync. Corrected the display name of the Platform "Everybody" group.
- Fixed an issue where the light mode collapsed toolbar showed the dark mode logo.
- Fixed an issue where the notification bell could show when there were no notifications.

- Fixed an issue where the Preserve SSH Client Process setting did not correctly display as checked.
- Fixed an issue where the SSH custom cipher was not applied when missing a value from the section.
- Fixed an issue where the synchronized groups displayed could sometimes return all the groups from the domain.
- Fixed an issue where the web launcher would not respect the mapped URL field when multiple URL fields existed on the secret.
- Fixed an issue where unnecessary audits could be written. Fixed an issue where DR Secret Server instances were ignoring licensing updates from Cloud Manager.
- Fixed an issue where upgrade banner was always showing when auto-update was off. Now shows only if at least one engine is lower version than latest.
- Fixed an issue where users could click New Secret multiple times when also uploading files.
- Fixed and incorrect launcher edit field description.
- Fixed buttons that should be grayed out. Run RPC Now can no longer be run when RPC is disabled. Run heartbeat Now can no longer be run when heartbeat is disabled.
- Fixed dark mode IBM password tooltips and banner color-contrast issues.
- Fixed edge case bug if SSH Block Listing causes duplicate sessions that break SSH Proxy.
- Fixed error that could occur when creating a new folder with the folder panel minimized.
- Fixed inconsistent logs between source and replica on partial success. Fatal error is now persisted across the wire so the replica is aware that the source had a fatal error
- Fixed incorrect logging error in AuthenticateWithAdConsumer.
- Fixed issue in directory sync where a search result with an attribute containing an empty list could cause an error.
- Fixed issue where the upper right search bar would not always switch to the selected secret when a selected secret was on a tab other than the General tab.
- Fixed issue with a test script modal where reopening the modal would show the selected secret's ID instead of its name.
- Fixed issue with folder permission editing when updating a path directly.
- Fixed link to dependency templates on the Secret Dependency tab.
- Fixed logic error where the RAS flag was not being referenced before deciding to delete the database entry that reflected additional users.
- Fixed long secret-template names to wrap better in folder edit.
- Fixed missing option. System group in Secret Server Cloud can now have metadata deleted.
- Fixed Platform permissions cached on Secret Server to replicate so they will be respected on a replica instance.
- Fixed query for obtaining services for a directory account in discovery. Fixed check on discovery source name when creating an empty discovery source.

- Fixed secret policies not showing as deleted after deleting a secret. Secret names on the RPC tab of a secret policy will now include "Inactive" if a secret is not active.
- Fixed text alignment. Left aligned the comment text on the MFA security view. The icon and button remain centered.
- Fixed the link to the subscription page from the banner.
- Fixed the REST API token endpoint path. The documentation generator, in removing the "api" string from the beginning of all routes, was also removing embedded occurrences. It now removes it only from the start of the route strings.
- Fixed the secrets grid on the Secret Erase Request Approval page (in a modal opened via a link button) that was obscured in dark mode and nearly indistinguishable in light mode. This is now an inline grid with auto-scroll.
- Fixed visual bug when removing current user's folder owner permissions.
- Folders in "Shared with Me" Quick Access menu are now filtered when searching.
- If a user's encrypted TOTP reset Guid gets corrupted, an administrator is now able to reset their TOTP.
- Improved error handling on the OpenId Configuration page.
- Improved the UI on the Collections Management page for advanced session recording agents.
- In the prior upgrade file set for 11.6.3, fixed an issue with SQL Delta 11.5.000006. Removed a SQL hint on the SQL index that was incompatible with non-Enterprise editions prior to SQL Server 2016 SP1 due to a compatibility issue with data compression. The incompatible hint was not necessary, so the delta was updated. Hashes for upgrade were updated for this change.
- Legacy RPC admin page removed.
- Legacy user and group management aspx pages removed.
- Limited Mode now goes to the correct link in SSC.
- Made performance improvements for the "What Secret Permissions Exist?" report.
- Prevented Thycotic One sync from syncg Platform Native users. This allows Platform native users to log in the rare situation they synced with Thycotic One. Then the administrator clears the system Platform User mappings.
- Queries executed in the chart and SQL editor for custom reports will now take the Use Database Paging setting into account so that the result is the same as if the query was being saved as a report.
- Removed legacy ASPX pages for secret templates.
- Removed link for managing licenses from the Cloud Subscriptions page.
- Secret Server was updated to use the same player for session recordings as platform.
- Set the GET SDK Client Account, SDK Client Audit, and SDK Client Rule API calls to set the operator parameter to 1 if it is not supplied by the caller when a User ID filter is specified.
- Switching pinned folders now resets the text search.
- Updated auditing for users modifying allowed cipher suite algorithms.
- Updated diagnostics page and licensing expiration checks to correctly handle non-US date patterns.
- Updated event subscription and workflow grids.

- Updated password requirement audits to correctly audit missed fields.
- Updated the action-handler secret-launch dialog layout to reflect design changes.
- Updated the Cloud Subscription page to the new UI.
- Updated the Dependency Changes List page to the new UI.
- Updated the Diagnostics page to the new UI.
- Updated the display for secret locked pages to address a wrapping issue with DoubleLock.
- Updated the distributed engine log UI updated. It now remembers your last selected site, system log grid UI updated, and the last selected log level.
- Updated the EventDetails token within Event Subscriptions to correctly capture secret comments.
- Updated the logout.aspx page to avoid errors being generated in rare cases when executing the SAML SLO flow.
- Updated the ticket system list page to the new UI.
- Updated user preferences page for better accessibility.
- web.config now allows explicit definition of allowed HTTP verbs.
- Addressed an issue where discovery rules would not correctly display the selected secret template or password type.
- Adjusted discovery scanning to minimize potential SQL deadlocks during the scanning process.
- Adjusted Secret Server and distributed engine to support 3.x versions of SAP .NET Connector.
- Converted HSM to a new UI page with a new PKCS11 API type. This new option enables you to protect your MEK and secret keys with an AES 256 key, bringing the strength of all keys to AES 256. After setting up PKCS#11 with your HSM vendor, you use the vendor cryptoki library (dll), token label and user pin to integrate with Secret Server. NOTE: You will need to disable the HSM first, to switch to the new PKCS11 API type. See the Hardware Security Module for more details.
- Enable Audit Integration on the Platform Configuration page can now be turned on.
- Extended timeout for some indexing steps for customers with over one million secrets.
- Fixed an issue that caused folder permissions to not update under specific circumstances.
- Fixed an issue where long column names did not wrap in the column selector.
- Fixed an issue where searching for a secret name using a substring within a single word would not always return results.
- Fixed an issue where the Dashboard Overview tab was not selected by default.
- Fixed an issue where the main search did not return content and updated the search design.
- Fixed an issue where the New Folder button would be incorrectly hidden in certain situations when displayed from Platform.
- Fixed an issue where the system log filter preference had an error when all was selected as the last used filter.

- Fixed an issue where the folder tree disappeared when there were more than 1,000 folders accessed and UAM was enabled.
- Fixed an issue with the discovery splash image margin.
- Fixed bug where changing the client ID did not update unless the client secret was updated as well.
- Fixed display issue for Secret edit modal on Discovery scope page.
- Fixed issue with QuantumLock Assign Users grid not displaying correctly after editing then canceling.
- Fixed the folder audit download to show the correct title.
- Improved exception logging for certain scenarios related to launching.
- License requirement message for secret policies updated to Pro Edition or higher.
- Removed no-longer-used bookmarklet login pages.
- Updated API documentation for updating team membership.
- Updated the Secret Import to handle a trailing whitespace in the folder path to prevent bug where created the child folder at the root level.
- Updated the ticket system detail page to the modern UI framework.

Secret Server Cloud Release Notes for September 23, 2023

Release Date and Notes

Cloud Release - All Regions Except United States of America (U.S) September 9, 2023

Cloud Release - United States of America (U.S) September 23, 2023

For changes pushed to cloud between major releases, see the Secret Server Change Log.

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.16

Protocol Handler: 6.0.3.26

Note: Step Upgrade Required (11.5.2). Versions prior to 11.5.2 will need to first upgrade to 11.5.2. The automatic downloads in the product will get the right versions for the step upgrade and then allow the 11.6 upgrade. But if offline and using the file upload method, versions prior to 11.5.2 will get an error message saying, "Integrity Check failed - Security Catalog is signed by thumbprint that is not specifically trusted." The remedy is to first upgrade to 11.5.2 and then do the upgrade to 11.6.

Feature Enhancements

View Own Session Recordings Permission

A new permission has been added "View Own Session Recordings". With this permission, a user can be restricted to only viewing the recordings that they initiated. If the user with this permission clicks on a recording initiated and owned by another user, they will get an Access Denied window. In addition, the "View Session Recordings" permission has been renamed to "View All Session Recordings".

Session Monitoring Playback Page UI Conversion

The Session Recording playback page has been updated to the new UI, including a new video player with additional controls. The SSH keystroke-only playback page and the video playback page have been merged, and the available elements will be shown. The legacy player is still available as a link from the new video playback page.

Note: The activity graph and download button have not yet been implemented for the new page, although they are available on the legacy page if needed.

Manual Password Change for Checked Out Secrets

Secrets with Change Password on Checkin configured now have the "Change Password Now" functionality available. This will enable the standard functionality of a password change, and the secret will also complete the automatic password change on checking in. This is to allow maintenance and testing of secrets protected in this manner, and a pending password change must be completed before the check-in process is allowed to begin in order to maintain a secure order of operations.

Updated User Selection Interface

Various locations around the product provide a user selection interface to provide the ability to select a user as the target of a particular configuration, such as the permissions, groups, and roles pages. These have been updated throughout the product to provide more data about the users in the list. You may now view, filter, and search for users by their Username, Display Name, or Email.

Automated Password Change on Import

An option has been added to the Import Secret function to mark each secret's password to immediately change after import. With this option enabled, a user who has had access to view the list of secrets will no longer know the password of the secrets once they have completed the import. The option is available for CSV and XML and can be flagged via the UI and API.

Enhancements

- Improvement: User tooltips in both Secret Server and Delinea Platform now highlight the Platform Integration Types.
- Improvement: (Disaster Recovery/Resilient Secrets):Data replication will now create personal folders for replicated users in cases where the replica blocks or does not allow personal folders to be replicated. This is only if personal folders are enabled on the replica.
- Improvement: User tooltips in both Secret Server and Delinea Platform now highlight the Platform Integration Types.
- Improvement: Fixed issues with user and group syncing between Secret Server Cloud and the Delinea Platform.
- Improvement: Added a "Managed" field to the Discovery Network view to show when a discovery item is managed.
- Improvement: The Password Requirement Audit has been converted to the new UI.

- Improvement: The Secret Dependency Changers editor has been converted to the new UI.
- Improvement: Dependency Templates are now available in the new UI.
- Improvement: Session playback player UI has been updated.
- Improvement: The Launcher Audits page has been migrated to the new UI.
- Improvement: Discovery Service Accounts Detail Page now shows services that run as the directory account as well as the computers on which that service runs
- Improvement: Added a Quick Access link to see all secrets you currently have checked out.
- Improvement: Updated Createuser.aspx to redirect to the new user creation page.
- Improvement: Updated the group role assignment UI.
- Improvement: Group membership assignment UI updated.
- Improvement: Group role assignment UI updated.
- Improvement: Updated process for populating a forthcoming computer-centric view.
- Improvement: Session recording search now uses updated filter pattern.
- Improvement: The built-in "Everyone" group was renamed "All Vault Users."
- Improvement: Enhanced new Discovery Area to include some additional fields and added logic for the error chip being displayed
- Improvement: Added a Copy button for Data Source URL on Disaster Recovery - Outgoing Setup Steps modal.
- Improvement: New Vault User Details in the Platform overview for Users tab. It requires a Vault to be successfully connected and configured for the details to appear, otherwise the section does not appear.
- Improvement: Added banners to various Roles/Permissions pages in Secret Server Cloud and Platform with links to help navigate between the two.
- Improvement: Secret Share tab UI has been updated to match the permission setting experience for setting folder permissions. Domain name is now displayed for users on the secret share tab.
- Improvement: Fixed an issue where the folder permissions tab would load slowly with large numbers of users.
- Improvement: Updated group membership management pages to use new design patterns.
- Improvement: The display name of the secret Vault is now set via the Platform. The Vault subcategories for Reporting, Inbox, and administration have been updated to reflect Secret Server.
- Improvement: Analysis tab of Discovery no longer includes disabled Discovery Sources in managed/unmanaged counts.
- Improvement: Administration Configuration Launcher Settings now displays the Enable Protocol Handler Auto-Update setting in cloud.
- Improvement: View Log was hidden for Directory Accounts since there's no computer associated to show the log of.
- Improvement: Added Application from tbAuditSecret to session search results model and session model.

- Improvement: When discovery is running the network view performance would timeout depending on SQL locks. This should no longer happen.
- Improvement: Discovery scanners added an option to "Add child scanner" which filters available scanners to show only applicable child scanners.
- Improvement: Disaster Recovery Add-On Licensing handling added.
- Improvement: Secret template fields table has been updated and has an improved drag and drop experience.
- Improvement: Secret panel is more mobile friendly.
- Improvement: Syslog/CEF logging enhanced to capture more detailed metadata for secrets that contain fields/data that map to the following SIEM fields: Account Name, Account Domain, Target Server, Request ID (that is from Ticketing System). Additionally, failed attempts to access secrets due to Ticket Validation errors are now also logged to Secret Audits.
- Improvement: New inbox notification bell with panel, allows for viewing and approving inbox items without having to navigate through the site.
- Improvement: The Security Audit Log page has been converted to the latest UI.
- Improvement: A donut chart showing different Operating Systems in discovery has been added to the Analysis tab of discovery.
- Improvement: Live viewing has been added to the new session monitoring.
- Improvement: The new UI Discovery Rules page now shows the correct Secret Template name.
- Improvement: Secret policy now links to the policy on the secret general tab.
- Improvement: A loading indicator now shows when opening the discovery add scanner dialog.
- Improvement: The main top left logo will link to the users preferred login home if it is the dashboard or all secrets.
- Improvement: The COM+ scanner will be able to be added, but there will be a note in the preview panel letting the user know that the scanner will not work for a site that is set to UseWebsite.
- Improvement: A preview chip has been added to Multifactor Authentication on Secrets and its supporting configuration pages.
- Improvement: A new field "Full Name" has been added to the discovery network view to give a more detailed version of the item's name.
- Improvement: Default columns have been added per Item Type in the discovery network view.
- Improvement: Dependency Tokens are now available on the dependency edit screen.
- Improvement: Enhanced loading times of Secret Server elements in Delinea Platform.
- Improvement: REST API documentation has links to individual services that load quickly.
- Improvement: Added filter on recorded-sessions endpoint to filter out applications, particularly 'RemoteAccessService' when in platform
- Improvement: Implemented a message shim in the Vault Broker to inform Secret Server that a user's platform permissions have changed.

- Improvement: Updated the Vault Settings and Vault User Detail Tabs with some UI changes.
- Improvement: Converted the creation of a Password Changer when Create Password Changer is selected from the Password Changers list in Remote Password Changing.
- Improvement: Added a filter of secretIDs to the Secret Search endpoint to that Secrets can be filtered by SecretID.
- Improvement: Terminate, limit to 5 minutes, and message only have been added to live viewing in the new session monitoring
- Improvement: The heading for Vault within Platform User Management details has been updated to read its value from within Platform.
- Improvement: The text for page title, breadcrumbs, and navigation for Secret Server Reporting have been updated in Platform to match.
- Improvement: Added Search Groups column to Discovery Network View.
- Improvement: Added more instructions regarding Disaster Recovery's data storage path configuration setting.
- Improvement: Added configuration setting to determine which secret permission is required to change Remote Password Changing settings on a Secret. Owner or Edit.
- Improvement: The breadcrumbs within the RPC administration pages have been standardized. The links within Platform Vault Configuration Overview no longer cause the page to reload.
- Improvement: Added a WMI Service Timeout setting to the cloud advanced configuration page to help with dependency changes that take more time than the allotted 60 seconds.
- Improvement: The PowerShell script timeout no longer defaults to 90 seconds. Instead, it now uses the value from the Event Pipelines Maximum Script Run Time (Minutes) setting in advanced configuration.
- Improvement: Improved performance of Secret Search for customers with large numbers of Secrets.
- Improvement: Updated data type to support frequent users of session recording that was crashing the encoding process.
- Improvement: Secrets with text field based URL lists are now searchable.
- Improvement: Platform users can login to Terminal using SSH Key Integration.
- Improvement: Added validation messages to password requirement rules for when password requirements are too complex to reliably generate a password.
- Improvement: When Platform integration is active the integration page will now have a button to reset mappings from Delinea Platform.
- Improvement: AD Privilege Password changer now has Remote Password Change timeout minutes Advanced Setting.
- Improvement: Better handling of unexpected heartbeat behavior to mitigate reported Distributed Engine stalling.
- Improvement: Connect As Credentials on Secret works better with SSH Keys for su user switching.
- Improvement: Updated links on the Security Hardening Report to new UI pages
- Improvement: When creating a new send to syslog task you no longer get a default schedule. Most of the templates didn't create a schedule, now they're all consistent.

- Improvement: Session monitoring search now supports searching by a single secret.
- Improvement: When a Secret is assigned to a site the user does not have access to due to Teams restriction, they will see the word "Restricted" instead of "Site Name (Inactive)".
- Improvement: Mitigated issue in large bulk secret actions.
- Improvements: The "Synchronization Running" message for DR will now only appear if there is a recorded start time for DR in the past and a finish time that is in the future.
- Improvement: Added Secret Field validation on the Template level to ensure users cannot create a "Secret Name" field on a template.
- Improvement: Default values for Secret Fields such as port will now be replicated for Disaster Recovery.
- Improvement: A user with only direct access to a report and the "browse reports" role permission can add that report to the dashboard.
- Improvement: The breadcrumbs within the RPC administration pages have been standardized. The links within Platform Vault Configuration Overview no longer cause the page to reload.
- Improvement: Report column preferences will be saved and applied when viewing a report.
- Improvement: The Secrets grid now updates displayed data and selected columns simultaneously.
- Improvement: Improved error logging and efficiency for calls coming from Delinea Platform.
- Improvement: Quick access filters now both apply when updated.
- Improvement: Knowledge base links within Platform Vault now link to their intended location.
- Improvement: Corrected edge case that could result in a session view audit being placed on the incorrect Secret.
- Improvement: The Parent Scan Template will be filtered to the type and will default to the first item in the list on create. The proper fields will be shown based on the type.
- Improvement: If a secret is inactivated after initially viewing the secret, a user that cannot view inactive secrets will no longer get an error from secret heartbeat.
- Improvement: Clicking cancel when editing folder permissions will clear any active filters.
- Improvement: Editing folder permissions now has a split button that allows for directly entering edit or add group/user mode.
- Improvement: The Secrets Quick Access link when collapsed, now targets the correct destination.
- Improvement: The Platform Opt In modal styling has been adjusted to no longer display with scroll bars.
- Improvement: Secret Share and Folder Permissions: Show disabled edit button until filters are loaded since split button does not yet support disabled.
- Improvement: API calls to /v[1/2]/secrets/{id} now update the Recents secrets data source.
- Improvement: When viewing Event Pipeline Activity details, selecting an Activity Detail record from the grid now displays the selected Activity's details.
- Improvement: Added query parameter for PipelineId to pass back when viewing specific pipeline activity
- Improvement: Minimum Heartbeat interval reduced from 15 to 5 minutes.

- Improvement: Discovery Scanner will not allow deletion until Secret selection is changed.
- Improvement: Remote Password Changing: Check for DNS Mismatch now visible and functional in Cloud.
- Improvement: EventTime token is available in pipeline scripts. $EventTime - event date and time of the event ("yyyy'-'MM'-'dd'T'HH':'mm':'ss").
- Improvement: The preview chips for Multifactor on Secrets have been removed.
- Improvement: Creating a User SSH Key in Platform downloads the private key with a proper filename.
- Improvement: Cipher Suite Configuration now allows configuration of allowed Host Key Algorithms.

Bug Fixes

- Fixed an issue where Viewing Session Connector Custom Launchers without access to the RDS Credentials secret would show an error.
- Fixed an issue where unplayable session recording videos would display an infinite load instead of the appropriate error.
- Fixed an issue where a secret template could be saved without RPC mappings configured.
- Fixed an issue where Pause times for ODBC Remote Password Changers were not adhered to. If you feel your RPC's are running slowly, check the pause times and remove them if they are not needed for the RPC action.
- Fixed an issue where Web Password Filler didn't work in certain instances due to an ambiguity in interpreting the Secret Server URL.
- Fixed an issue where setting custom expiration dates in all time zones did not work correctly.
- Fixed an issue where all event subscriptions did not fire for secrets in subfolders of the target folder.
- Fixed an issue where the secret name would incorrectly display on the New Discovery Import Rules page.
- Fixed an issue with negative numbers exporting incorrectly when exporting to a CSV file.
- Fixed an issue where a large number of SSH terminal connection history records causing timeouts.
- Fixed an issue with hidden days until deletion field when enabling deletion in the retention schedule. Added localization to error when trying to submit days less than or equal to the archive retention value.
- Fixed an issue with passwords being uneditable if RPC is set to use a Privileged Secret to which the user has no access to. Restored explanatory banner.
- Fixed an issue where secrets aren't synced with DevOps in cloud with when triggered by pipelines.
- Fixed issue in discovery where computer scans were sometimes throwing string truncation exceptions.
- Fixed an issue where TOTP Secret Settings edit button was available to users who could not edit the settings.
- Fix an issue with editing Session Connector Custom Launcher Port.
- Fixed a UI issue with the launcher popup window showing an option the user didn't have permission for.
- Fixed an issue where configuring a new session connector launcher might not show all available launcher types.
- Fixed an issue where configuring "Use Additional Prompt" on launchers might prevent save.
- Fixed an issue with the TemplateCreateSecret role link.

- Fixed an issue with View Launcher Password.
- Fixed an issue where users with only 'View' access on a Secret would be unable to view the Password if there was a custom launcher with arguments configured for that Secret Template.
- Fixed an issue with DSV sync for secret with file type fields and no file set.
- Fixed an issue with localization on folder Metadata page.
- Fixed an issue with sorting for Checkout User Id and Checkout User.
- Fixed an issue with ODBC password changing that broke postgres and mySQL changing.
- Fixed a logging issue with Dependency changes ran through Distributed Engine being skipped due to conditions.
- Fixed an issue where the generate SSH key returns a 500 exception.
- Fixed an issue where the SSHCipherSuiteModel GetAsync returned a 500 exception.
- Fixed an issue where the CreatePublicSSHKey returned a 500 Exception.
- Fixed an issue where Discovery Scanners could not be removed until the associated secrets had been edited.

Secret Server Archive

This section contains release notes for unsupported versions of Secret Server On-Premises and older versions of Secret Server Cloud.

For changes pushed to cloud between major releases, see the Secret Server Change Log.

On-Premises Archived

This section contains unsupported versions of Secret Server On-Premises.

Secret Server: 11.3.000003 Release Notes

11.3.000004 is a retroactive patch for this release. Please see "Secret Server 11.7.000001 Release Notes" on page 1596 for details.

Important security release–we recommend all affected Secret Server on-premise customers upgrade as soon as possible.

Release Date: February 8, 2023

An unauthorized access vulnerability was found in Secret Server. This issue is rated **High** with an 8.5 Common Vulnerability Scoring System (CVSS) score. Please see the CVSS Calculator for details.

11.3.000003 is a security update only.

This vulnerability has been patched in Secret Server Cloud, so there is no additional update to address it.

Secret Server: 11.3.000002 Release Notes

Release Dates and Notes

On-Premises: November 15, 2022

Cloud: November 11, 2022

Component Versions

Distributed Engine: 8.3.1.0

Protocol Handler: 6.0.3.23

New Features

None

Enhancements

Discovery

- Improved discovery dependency matching performance.

Secrets, Policies, and Templates

- Added a secret policy setting to control "Change Password Upon Check-In" behavior. Previously, this was enabled if "Require Check Out" was enabled.
- Optimized Performance of /api/v2/secrets endpoint when searching by secret name.

Session Recording and Monitoring

- Added a terminate option to the session playback page

Bugs

Access Requests, Checkout, Secret Workflows, and Doublelocks

- Check in now works as expected if settings "Check In Secret on Launcher Close" and "Close Launcher on Check In Secret" are both false.

Alerts, Auditing, and Logs

- Fixed an error in SAML audits.

Authentication, Login, and Directory Services

- Fixed an issue where deleted Active Directory groups would not be reflected as disabled during Active Directory synchronization. This now correctly registers as a disabled group and is disabled in Secret Server if configuration is set to mirror the AD status.
- Fix an issue with RADIUS challenges in Secret Server Cloud.
- Fixed an issue where installation on specific dates on servers with a dd/mm/YYYY localization configuration would prevent reading some configuration settings.

Encryption, Passwords, and Certificates

- Fixed an issue where longer passwords with spatial pattern complexity rules enabled would fail to generate. Spatial patterns are now compared at a length of four characters if the password is above 50 characters.

General

- Fixed an issue where the Secret Server website would not load if the internal site connector is unavailable at startup.

Remote Password Changing

- Added the User parameter to the IBM iSeries Mainframe Connection for launching, password changing and heartbeat functionality.
- Fixed an issue where PowerShell-based dependency changers would not correctly pass arguments.
- Fixed an issue where removing an RPC schedule from a secret policy does not update a secret assigned to the policy.
- Fixed an issue where the IBM iSeries password changer was not properly adding the model value to the connection string.

Secrets, Policies, and Templates

- Fixed an error when converting a secret in a folder with a launcher settings policy.
- Fixed an issue where the test buttons would not function for the Oracle Account Ver. 2 password changer.
- Fixed an issue where "Days Until Expiration" value on the secrets grid would show a large negative number if

expiration is forced. This now displays "Expiration forced."

- Fixed an issue where sorting by folder path in the secret grid view would return an error.

Session Recording and Monitoring

- Fixed an issue where the advanced session recording agent would attempt to make many reconnections in a short time span.
- Fixed an issue where monitoring and termination of live sessions was not appearing in the New UI. This now takes users to the regular session playback page, which displays the live session.
- Fixed an issue where file contents during SFTP/SCP file transfers were not included in session keystroke recordings.
- Fixed an issue where the new UI session monitoring search required additional permissions to load.

Future and Recent Deprecations

Note: This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server: 11.3.000001 Release Notes (GA)

Release Dates and Notes

Early Access: October 4, 2022 (On-Premises)

General Availability: October 25, 2022 (On-Premises)

Note: The following release notes apply to both EA (11.3.000000) and GA (11.3.000001). The bug fixes that only apply to GA are prefaced with *GA only*.

Note: The protocol handler version for this release is RDPWin_6_0_3_23.

New Features

Additional Custom Logo Variations

You can now upload more custom interface logos. This includes an optional larger logo for the login page. Each logo includes alternative images for light and dark modes. This results in six logo variations.

The settings are available in the configuration preview. Enable the configuration preview and then locate the new settings in the **Administration > Configuration > General > User Interface** area.

Advanced Session Recording Agent

The advanced session recording agent (ASRA) MSI is now self-contained and based on .NET Core, which does not need a .NET SDK installed; however, installing an ASRA from the advanced session recording page using the installer still requires that the .NET Framework redistributable components are installed.

To install or deploy the agent without the framework, follow the customization steps here: "Installing the Advanced Session-Recording Agent" on page 1240 .

Configurable Global Banner

You can now configure a multipurpose global banner for all users. You can use it for maintenance, security, or policy notifications. You can set the severity level, text, a hyperlink, and an in-theme color, which is determined by the severity.

The settings are available in the configuration preview. Enable the configuration preview and then locate the new settings in the **Administration > Configuration > General > User Interface** area.

Classic User Interface

We enhanced the new UI experience, and all new features only appear in that UI. The classic UI is deprecated and no longer maintained. For this release, the classic UI is disabled by default but temporarily still available. You can enable it at **Administration > Configuration > General > User Interface settings.** As of the next minor release, the classic UI will be permanently removed.

Disaster Recovery Synchronization Improvements

We improved the disaster recovery (DR) feature to synchronize additional items, including:

- Local and domain users
- Groups
- File attachments
- Secret and folder permissions

This enables replication of the permissions structure to the replica server, providing a readily available standby with permissions in place.

Event Pipeline Enhancements to Sending Emails

Users can now select an inbox email template in the event-pipeline send-email tasks. This gives the user access to event-pipeline and inbox tokens within the predefined email template.

Once the event pipeline is triggered, it sends an inbox notification and processes any inbox rules. The inbox rules send an email to the task recipients.

Protocol Handler Process Tracking

We renamed launcher settings from "Record Multiple Windows" and "Record Additional Processes" to "Track Multiple Windows" and "Track Additional Processes." These are now both used for process tracking, regardless of whether recording is enabled. This improves session termination, either triggered or automatic, applying to child and specified processes, per the launcher settings.

Refreshed Administration Page

We updated the administration page with a new categorized view where users can pin commonly used items to a list. To pin an item on the new administration page, hover over or focus it, revealing a pin icon. Click the icon to add the item for quick access.

Session Monitoring Page

We converted the session monitoring search page to the new UI. The functionality remains essentially the same; however, we removed role restrictions on filters. We also optimized both the front- and back-ends.

SFTP Tunneling

Added an SFTP tunnel setting to the SSH proxied process launcher for use with SFTP client custom launchers. This was tested with FileZilla and WinSCP.

SSH Cipher Suite Configuration

We added a configuration page that sets the Secret Server SSH ciphers used when making SSH connections for various tasks, such as heartbeat, password changing and discovery. This does not apply to SSH password changers using the "Legacy" runner type. This is at **Administration > SSH Cipher Suite Configuration**.

With this feature, users can set availability and application order for key exchange, MAC, and encryption algorithms via an easy-to-use list. To use the list, go to the configuration page for the site, and enable the SSH cipher suite setting.

Enhancements

Alerts, Auditing, and Logs

- Better error logging during disaster recovery.
- Added a "Session Recording Downloaded" audit event.
- Updated email templates with new brand headers.
- Added a saved history for TOTP fields on secrets.
- Converted the user audit report page to the new UI.
- Added support for Unicode characters in email addresses.

Authentication, Login, and Directory Services

- Renamed the "What computers in Active Directory no longer exist?" report to "What Computer Accounts found by Discovery are not managed?" to reflect the results more accurately.

Backup, DR, and HA

- Added file attachment fields to disaster recovery sync.
- GA only: Improved replication by adding extended data mappings.

Dashboard and UI

- Searches now work correctly for usernames with non-alphanumeric characters such as % _ and [].
- Updated Inbox page to allow for low screen resolution or high zoom settings.

Encryption, Passwords, and Certificates

- Generated SSH private keys are now saved with AES128 encryption instead of DES.

General

- Updated several third-party libraries to remediate vulnerabilities.
- Added DPAPI and SAML pages to the configuration preview.
- Office 365 Template and Password Changer have been renamed to Azure AD Account and Azure AD respectively.

Heartbeats

- Office365 heartbeats now detects if you provide the domain in the username.

Launchers and Protocol Handlers

- We now include the required Visual C++ runtime with the protocol handler.
- Updated the PuTTY version distributed with the protocol handler from v0.74 to v0.77.
- Launcher mappings and restrictions are only validated when they have changed.

Localization

- Increased localization performance in Cloud by caching additional language files.

Remote Access and Proxies

- Added a "use SFTP tunnel setting" to the SSH proxied process launcher to support using SFTP client custom launchers. This was tested with FileZilla and WinSCP.

Reports

- Added "new report " and "add category " action buttons to the reports page.
- Moved reports audit to a tab for consistency, replacing the existing button.

Secrets, Policies, and Templates

- Added more tooltips to the secret template page.
- Added a new public endpoint for secret template conversion.
- Increased an individual secret list field to 500 characters.
- Secret erase now completes without errors when the secret has an associated list.

Session Recording

- Adjusted SSH keystroke recordings to not display VTY special commands to clean up the output and to show control commands used in the client output.
- Converted the advanced session recording agent to .NET Core.

- Advanced session recording agent will now only register if the latest agent is installed (or any future versions). This supports a change in the registration process.

Teams

- Added "include all users from domain" setting to the team configuration, which directly maps the users synchronized from a domain to that team.

Users, Groups, Roles, and Permissions

- Viewing the password requirement page now requires the "view password requirements " permission.

Web Password Filler

- Added a Safari link for Web Password Filler to the launcher tools page.

Bugs

Access Requests, Checkout, Secret Workflows, and Doublelocks

- Fixed an issue where no notification appeared if a secret view interval expires while the user was viewing the secret.
- Fixed the secret session extension to also extend launcher sessions and not just checkout sessions.
- Fixed an issue where adding a workflow to a secret could not be saved.
- Fixed an issue where users could not check out a secret when the internal site connector was unavailable, even if they did not have checkout hooks or pipelines actions that rely on the site connector.
- Fixed an issue where workflow items in secret policies would not save.

Alerts, Auditing, and Logs

- Fixed an issue where the secret-template password-type field-mapping audits were not recording the correct fields
- Fixed an issue where viewing a secret list would record that the password was displayed in the audit log.
- Fixed an issue where removing a user from a group did not show up in the event log.
- Fixed issue where "expire secrets on user audit report" was deactivating secrets.

API and Scripting

- Fixed an issue where secretPath in the API would not work with secrets having a single forward or back slash in their name.
- Fixed an issue where the diagnostics API showed an incorrect value for the upgradeAvailable property. We also added the latestVersion property.
- Fixed issue with secret search v1 API throwing a 400 error.
- Fixed issue where the secret policy SOAP API would throw an error when using the professional license.

Authentication, Login, and Directory Services

- Fixed an issue where SSH terminal authentication with RADIUS two-factor authentication would incorrectly audit login failure.
- Fixed an issue where SSH terminal authentication would not correctly process RADIUS logins.
- Fixed an issue where changing the distinguished name field on directory services displayed inaccurately in audit logs.

Background Services

- Fixed an issue where running a change-password bulk operation required an SSH key on secrets set up for both key and password changes.
- Fixed an issue where bulk operations would fail if too many target secrets were specified.
- Fixed an issue where a bulk change-password operation threw an "empty private key" error.
- Fixed an issue where the "set privileged account" bulk operation would display an error when attempting to set the "credentials on secret " option.
- Fixed an issue where applying a bulk operation checkout would also enable "change password on check In."

Backup, DR, and HA

- GA only: Fixed an issue where pre-existing non-replicated folder ACLs were deleted on replication.
- GA only: Fixed an issue where during DR ProtoObjectCollectionServiceBase.SaveDtos (secret launcher) threw a fatal error when too many characters were inserted.
- GA only: Fixed an issue where PasswordRequirementConsumer threw errors during replication when bringing in large numbers of secrets.

Bulk Operations

- Fixed an issue where running a change-password bulk operation required an SSH key on secrets set up for both key and password changes.
- Fixed an issue where bulk operations would fail if too many target secrets were specified.
- Fixed an issue where a bulk change-password operation threw an "empty private key" error.
- Fixed an issue where the "set privileged account" bulk operation would display an error when attempting to set the "credentials on secret " option.
- Fixed an issue where applying a bulk operation checkout would also enable "change password on check In."

Cloud

- Fixed an issue where cloud pages were returned in a previously cached language.

Dashboard and UI

- Fixed an issue where changing the address bar from one secret ID to another would not update the display of the associated secrets list.
- Fixed an issue where the "Restrict SSH Commands Edit" link was not showing.
- Fixed an issue where the password on a secret was not displayed when "Edit Requires Owner" was set on the field and the user had edit permissions.
- Fixed an issue where the script editor on various pages would act as a keyboard trap, and there was no way to change focus without a mouse. Enabled an escape combination and detailed it alongside the editor.
- Fixed an issue where the script editor on various pages would represent a tab with a single space. Changed to 4 spaces.
- Fixed an issue where text formatting was not displaying on the in-line secret preview in the secret grid.
- Fixed an issue where breadcrumbs were not displaying on several pages.
- Fixed an issue where custom Logos would not display correctly.
- Fixed an issue where a long secret field name would overflow the label area and obscure the value in the secret preview panel.
- Fixed an issue where page layout would change the location of buttons on the password changer configuration page.
- Fixed an issue where setting the UI inactivity timeout value to 0 caused an exception. This value now disables UI inactivity as expected.
- Fixed an issue where the secret list view would not scroll to the bottom of the list.
- Fixed an issue where secrets view would not remember the last setting between grid or list options.

Discovery

- Fixed an issue where you could not set a discovery source to inactive if the sync secret was no longer valid.
- Fixed an issue where discovery rules would not function correctly when organization units had brackets in their names.
- Fixed an issue where secret search filters were not saved to a discovery scanner when searching "All Folders."

Distributed Engines and Site Connectors

- Fixed an issue where a site connector could not be disabled if it was assigned to the local site when it is configured for website processing.

Encryption, Passwords, and Certificates

- GA only: Fixed an MEK rotation error when rotating deleted session recordings.
- Fixed an issue that could cause an index error during master key rotation.
- Fixed an issue where a password rules validation message did not appear when exceeding the maximum

allowed characters for a ruleset.

- Fixed an issue where some validator options were not visible when creating a new password requirement.

Event Subscriptions and Pipelines

- Fixed issue where users could not click pipeline buttons when in unlimited admin mode.
- Fixed an issue where the "Move to Folder" event pipeline task would throw an error when editing.
- Fixed an issue where event subscriptions could only target secrets with view permission or higher. Secrets with list view are now selectable.

Folders

- Fixed issue that allowed sharing personal folders when using "Username and Domain" as the display name.
- Fixed an issue where a new pinned folder view would default to all secrets instead of the expected default active secrets.
- Fixed an issue where folders would unintentionally display in reverse alphabetical order in the secrets list.
- Fixed an issue where expanding the details of a folder would sometimes display the details for a secret in the secrets grid view.
- Fixed an issue where the folder picker would not display when selecting from a list of collapsed folders in secret list view. When there are many subfolders, they collapse into a single item showing the number of subfolders. Clicking this opens a folder picker to choose the desired folder.
- Fixed an issue where the last viewed secret folder was not retained when the user preference was set to remember it.
- Fixed an issue where secret policy inheritance would not propagate down the folder structure when importing folders.
- Fixed an issue where moving a child folder to a location with a subfolder of the same name would throw an error.

General

- Fixed an issue where Secret Server On-Premises unsuccessfully attempted to connect to our cloud monitoring platform.
- Fixed an issue where dependency scan messages would not expire when not consumed, causing a large queue to form.

Heartbeats

- Fixed an issue where heartbeats intermittently failed when run while storing account details for multiple Office 365 tenants.
- Fixed an issue where heartbeat could be queued while a remote password change operation was still in progress. This would cause the heartbeat to fail.
- Fixed an issue where Oracle usernames with the hyphen character would fail to heartbeat. This now wraps the username in quotes, as is required by Oracle.

Import and Export

- Fixed an issue where automatic export would not run every day as configured.
- Fixed an issue where secret export did not trigger the appropriate event subscriptions or pipelines.
- Fixed an issue where extended fields in the secrets view grid would not be present in the export.

Launchers and Protocol Handlers

- Fixed an issue where the protocol handler for session connector download would fail.

Licensing and Activation

- GA only: Fixed an error during license activation.

Localization

- Fixed an issue with slow page loading when the browser language was set to a language that is not an available localization option.
- Fixed an issue with localization not applying to some labels on inbox notifications.

Remote Access and Proxies

- GA only: Fixed an issue with SSH proxy where WinSCP in SCP mode with block lists did not connect as expected. In addition, the session recording log no longer records file content.
- Fixed an issue where system tray notifications would not show in session connector sessions.
- Fixed an issue where duplicate jumpbox route options were sometimes shown.

Remote Password Changing

- Fixed an issue where custom PowerShell password changers would not evaluate the $CURRENTPASSWORD token.
- Fixed an issue where changing the remote password changing schedule by policy would not happen until after the next previously configured schedule attempt. This now immediately takes effect.
- Fixed an issue where monthly remote password changing schedules could not be saved.
- Fixed an issue where remote password changing schedules would display time in an incorrect time zone.

Reports

- Fixed an issue where built-in remote password changing reports were excluding items without configured expiry fields. Added a new field to the secret table to track the last password change and updated the reports.
- IBM Only: Fixed an issue where the report editor was not usable.
- Fixed an issue where report results could not be downloaded from the view report page.

Secrets, Policies, and Templates

- GA only: Fixed an error where rendering secret list pages threw a SQL error.
- GA only: Fixed an error where a user could create a secret using a secret template the user does not have access to.
- GA only: Fixed an issue where the secrets reporting card description was missing.
- Fixed an issue so only secret owners can see the test option for a dependency.
- Fixed an issue where service dependency changers failed because Windows used a different naming format from the secret. We added more-robust handling for several cases.
- Fixed an issue where an imported dependency would persist even if the machine no longer existed.
- Fixed an issue where a user could change the active state of a secret template when not authorized to do so.
- Fixed an issue where you could save a secret with no owner set.
- Fixed an issue where setting permissions on a secret would result in an error.
- Fixed an issue where converting a secret to a template would remove metadata.
- Fixed an issue where an error occurred when changing the autochange schedule on a policy from default to enforced.

Session Recording and Monitoring

- GA only: Fixed an issue where session monitoring did not save a column preference.
- Fixed an issue with the advanced session recording agent installer generation.
- Fixed an issue where large numbers of recorded keystrokes would affect performance in the secret audit page.

Ticketing System

- Added better logging and error handling for ticket system integrations.

Users, Groups, Roles, and Permissions

- Notification: System roles are no longer editable. If you need to enable a system role, please contact Delinea Support.
- Fixed an issue where the inactive message would show when a user had view only permission on a secret.
- Fixed an issue where users could not access the secret settings page without the "view advanced secret settings" role permission.
- Fixed an issue where users could not define a domain when creating a user.
- Fixed an issue where only 30 of the user management IP restrictions would load.
- Updated secret share to not show users without permission on the secret.

Web Password Filler

- Fixed an issue where forward slashes were removed from Web Password Filler URLs. This broke redirect URLs.

Future and Recent Deprecations

Note: This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server: 11.3.000000 Release Notes (EA)

Important: EA release notes are combined with the "Secret Server: 11.3.000001 Release Notes (GA)" on page 1713.

Release Dates and Notes

Early Access:

October 4, 2022 (On-Premises)

General Availability:

October 22, 2022

Note: Technically, the release number for 11.3 EA is 11.2.000050. GA will have the correct numbering.

Secret Server: 11.2.000003 Release Notes

11.2.000004 is a retroactive patch for this release. Please see "Secret Server 11.7.000001 Release Notes" on page 1596 for details.

Important security release—we recommend all affected Secret Server on-premise customers upgrade as soon as possible.

Release Date: October 4, 2022

A SQL Injection vulnerability was found in the Secret Server REST API. This issue is rated **High** with a 8.8 Common Vulnerability Scoring System (CVSS) score. Please see the CVSS Calculator for details.

11.2.000003 is a security update only.

This vulnerability has been patched in Secret Server Cloud, so there is no additional update to address it.

Secret Server: 11.2.000002 Release Notes (GA)

Release dates:

General Availability:

June 7, 2022 (On-Premises)

June 11, 2022 (Cloud)

Note: The following release notes apply to both EA (11.2.000000) and GA. The bug fixes that only apply to GA are prefaced with *(GA only)*.

Note: The known issue with the replication process of the new disaster recovery feature launched in 11.2.000002 (June 7th, 2022) has been fixed.

New Features and Enhancements

New Branding

Thycotic and Centrify are now Delinea, and we have updated the interface styling to reflect the new company branding. Both light and dark mode are updated with new colors and component styles.

Improved Secret Browsing Experience

Note: The classic UI was scheduled for removal in this release. Based on feedback, we deferred it to a later release. This section details the changes made to address the feedback on this change.

The secrets menu item in the new UI now has a collapsible folder tree panel that occupies the full height of the page. This replaces the folder tree at the bottom of the left navigation bar and delivers a significantly improved folder browsing experience.

Clicking a secret in the secret table view now opens an in-line card with details and actions. A launch button now appears when hovering over the row, where launchers are available, allowing launching without viewing secret details.

You can now pin a folder, limiting the folder tree to showing the contents of that pinned folder. With this, you can limit navigation to the folder and streamline navigation. Applying filters to the table in one pinned folder will not affect others, and the settings are remembered upon returning to that pinned folder.

We moved "favorites," "recent secrets," and "shared with me" to a widget bar, globally available in the top right in the new UI, allowing immediate access to your most important items.

New Secret Policy UI

We converted secret policy configuration to the new interface, adding a few changes. Policy settings are now grouped into tabs, grouping related settings together.

The "enforced" or "default" dropdowns are now checkboxes, improving usability.

New Administration Side Panel

A new administration menu is now available as a global side panel, allowing quick access to administration pages through a searchable list.

You can also pin pages in the list, creating a tailored quick-access list specific to a user. If the pinned item list is empty, there is a button to populate it with the default items that appeared on the previous list, which was available in the bottom-left "Admin" menu.

New Configuration Preview

A new configuration page is almost ready, and we need feedback on the new experience before making the page live. There is a new link to enable it in the top right-hand corner of the current configuration page.

Enabling the configuration preview changes the behavior of the new left administration panel, mentioned above, to include a new configuration section where the subsections are listed. Accessing any of these subsections navigates to the new configuration page, which has a dedicated search. This searches for configuration items by label, tooltip, and value content, displaying the search results and providing a link to a configuration section.

To disable the configuration preview, browse to the new configuration page, and disable the configuration preview in the top right-hand corner.

New Disaster Recovery Replication

There is now an efficient data replication method between multiple instances of Secret Server. This is typically used for replicating secret data into a backup vault for disaster recovery or business continuity. Currently, this is limited to secrets, templates, and launchers. Currently, DR replication does not include any other data, such as permissions, users, or policies, which future Secret Server versions will likely have.

Important: There is a known issue with the replication process of the new disaster recovery feature launched in 11.2.000002 (June 7th, 2022). A resolution is in progress and will be available soon. Please delay testing or implementing this feature until we release an update.

Note: DR replication is available in Secret Server Platinum.

New Oracle Password Changer

A new Oracle password changer is available with new templates pre-configured for various configurations. The new password changer does not require installing any additional components on web servers and distributed engines, unlike the existing changer, which required Oracle ODAC components.

The new changer supports Oracle AS SYS, Oracle DataSource, and Oracle TCPS connections, and the new templates are pre-configured for these.

New Marking and Obfuscating PII

There is a new option available to enable the marking or obfuscation of Personally Identifiable Information (PII) in audit exports. This allows for data exportation for review by third parties without including any PII. Marking PII prepares exports for external cleanup, and obfuscation automatically removes it during exportation.

PII includes many internal stored attributes, such as IP addresses, usernames, and email addresses. Metadata fields can be flagged on creation as potentially containing PII, which aids applying the same filtering to user-configured metadata fields.

This feature currently only applies to audit tables available in the interface.

New Password Complexity Rules

We have added two new optional features to password requirement configuration:

- Variable Rule Matching: Passwords must match a specific number of your password character sets. Previously, all defined character sets must match. Now, you can define that one or more sets must match. For example, three of four sets must match.
- Must End With: A password's final character must match a defined character set. This support compatibility with systems with similar requirements.

New Secret Checkout Expiration Notification

There is now a globally configurable checkout expiration that notifies users with a checked-out secret via their Secret Server inbox and inbox rules when a defined checkout period percentage has elapsed. For example, setting checkout expiration to 80% notifies users when 20% of the checkout interval remains.

This setting is available under "user experience" in the configuration page.

New Role-based Access Control for Reports

We have added a "Browse Reports" role that allows access to reports restricted by permissions. Permissions are configurable at the category and report levels and share a similar inheritance model to secrets and folders. You can define users or groups with "view" or "edit" permissions for each category or report.

Users with the existing "view reports" and "edit reports" roles are not restricted by the permissions set.

New Extendable Access Requests

You can now request future access to a secret you currently have access to—no need to wait for the current access to expire before asking for more time. A new access window is granted using the existing approval method. The new window's start time is automatically set to the end time of the current one.

General

- Secret exports now support using a doublelock password instead of a local password, providing a method for federated users to manage passwords.
- RDP Settings, such as accessing the clipboard, are now enforceable on secrets by policy.
- Web Password Filler now supports URL lists. You can map a list to the URL field in web launcher mappings.
- Restored support for AES-*-ctr algorithms in Secure Blackbox when running in FIPS mode. SSH heartbeat, RPC, and proxy can once again connect to machines using –ctr algorithms.
- Added new tokens for recipient time, server time, and UTC time to inbox templates.
- Syslog messages are now sent to SIEM systems in UTF-8 encoding, which supports non-ASCII characters. Users who wish to view UTF-8 characters, such as Cyrillic or Korean, in their SIEM system will need to configure their SIEM servers to receive UTF-8 messages.
- Updated the protocol handler for session connector.
- Optimized SQL queries around RPC scheduling to increase performance.
- Web launcher support for HTTP: Web launchers by default force HTTPS, regardless of the protocol scheme in the URL field. Advanced configuration now as a "Web Launcher Forces HTTPS" setting to disable this for http:// URLs.
- Optimized performance of APIs related to Web Password Filler.

- Added a "preserve client process" setting to SSH-proxied process launchers that preserves the session if the launched process closes. This allows tabbed SSH clients to operate correctly as launched sessions.

UI/UX

- Log in autocomplete is now always on. We deleted the "allow autocomplete" setting from Admin > Configuration > Login.
- SSH Terminal now displays the last heartbeat status of a secret in the output of the "cat" command.

API

- Created a more usable version of the secret policy API.
- Adjusted /api/v1/folders sort order to return deterministic results.

Bug Fixes

Access Requests, Secret Workflows, and Doublelocks

- Fixed an issue where extend checkout would fail for users with view permission.
- Fixed an issue that did not allow custom check out intervals when set to maximum values.
- Fixed an issue with the Revoke button being invisible in the approval and requests inbox. It is now visible for basic approvals but not for workflow approvals as these do not support revocation.
- Fixed an issue where doublelock access would be intermittent if requests hit different web nodes.
- Fixed an issue where a workflow could be created and activated with no approvers

API and Scripting

- Fixed an issue where an empty "items" property existed on the api/v1/lists/{categorizedListid} endpoint.
- Fixed an issue where applying a policy with "Web launcher requires Incognito mode" set during API secret creation caused creation failure.
- Fixed an issue where an empty "items" property existed on the api/v1/lists/{categorizedListid}.
- Fixed an issue where JSON values could not be used as script arguments. These are now handled correctly in PowerShell.
- Fixed a SQL timeout issue from the adjusted data retention cleanup of the SDK client table.
- Fixed an issue where duplicate metadata fields could be created via the API.
- Fixed an issue with the DomainStatus value not being correctly populated in calls to the GET /api/v1/directory-services/synchronization endpoint.
- Fixed an issue where the /api/v1/directory-services/synchronization/log endpoint would not return data. This endpoint was renamed to /logs and now returns appropriate data.

Alerts, Auditing, and Logs

- Fixed an issue where an IP address was not displaying in the user audit page in the new UI.
- Fixed an issue where incorrect messaging was logged due to an IP address change. The log no longer references user location.
- Fixed bug where a "value cannot be null." error was written to the logs when updating node records during upgrade.
- Fixed an issue with URL generation for inbox notification emails.
- Fixed an issue where an incorrect IP address could be sent to syslog.
- Fixed an issue where verbose logging in distributed engines that indicates which key it is encrypting and decrypting communications with Secret Server specified it is logging a hash of the key, not the key itself.
- Fixed an issue where operational logs for multiple processes would not clean up due to the database table being too large.
- Fixed an issue where dual controls would not insert correct audits for launch events.

Authentication, Login, and Directory Services

- Fixed an issue where privileged password changing would not work for a user defined by the user principal name. The Active Directory password change now works with both UPN and SAM account name formats.
- Fixed an issue where SAML logins would not reset login failures for account lockout policies.
- Fixed an issue where OpenLDAP directory synchronization would not accept a custom port.
- Fixed an issue where the login page would load slowly when using HSM integration.
- Fixed an issue where CAPTCHA would sometimes fail due to session persistence.
- Fixed an issue with SAML SLO failing when the identity provider failed to log out.
- Fixed an issue where an incorrect identifying IP address was sent to Duo. This now sends the client address.
- Fixed an issue where Azure AD directory synchronization failed due to an unexpected value for OnPremisesImmutableId. These values are now anticipated.
- Fixed an issue where SAML login could be attempted for users assigned to disabled domains.
- Fixed an issue where the download service-provider metadata in SAML configurations showed even though there was not any metadata to download.
- Fixed an issue where engine logs would be cleaned up in a single transaction that could time out.
- Fixed an issue where Azure AD synchronization would incorrectly map users that had their username changed.

Disaster Recovery

Note: The known issue with the replication process of the new disaster recovery feature launched in 11.2.000002 (June 7th, 2022) has been fixed.

- (GA only) Fixed an issue with disaster recovery synchronization not updating synchronized folder names.
- (GA only) Fixed an issue with disaster recovery synchronization not synchronizing secrets predating a new synchronization folder.
- (GA+ only) Fixed an issue where the CLR detected an invalid program.
- (GA+ only) Fixed an edge case issue where data may fail to replicate if edits occur while a replication is taking place.
- (GA+ only) Fixed an edge case issue where editing a secret field launcher may cause a null reference during DR replication.
- (GA+ only) Fixed an InvalidCastException that was introduced in the fix for RC13.
- (GA+ only) Fixed an issue where launcher sessions expired immediately when launched from the DR replica.

Discovery

- Fixed an issue where changing the secret on a discovery import rule would not save.
- (GA only) Fixed an issue where GCP discovery would return no results and not create the source organizational structure if errors to certain calls were received.

Encryption, Passwords, and Certificates

- Fixed an issue where credentials were not correctly supplied to "post change failure" or "post change success" commands, resulting in a failed connection instead of these commands running.

Event Subscriptions and Pipelines

- Fixed an issue where a secret metadata filter in an event pipeline would not work unless another task or filter had a $ token.
- Fixed the language resource for event pipeline policies.
- Fixed an issue where group name was not present when an event subscription was created for "User - Removed from group."
- Fixed an issue where subscribers could not be removed from an event subscription notification rule.
- (GA only) Fixed an issue with event pipelines not running with certain date formats.

Folders

- Optimized folder tree loading when over 1000 folders are displayed.
- Fixed an issue where "delete folder" was not available in the row options menu for folder items.
- (GA only) Fixed an issue where folder edit view would not update when editing one folder following another.

General

- Fixed an issue with custom dictionaries attached to password requirements not updating correctly.
- Fixed an issue where a problem with a single site caused messages to stop processing for multiple sites.
- Fixed an issue where the session connector would not install on non-English language systems.

- Fixed a CVE-2018-1185 vulnerability in the log4net package, which was updated to 2.0.14.
- Updated the time zone database library with the latest information.
- Fixed the Connection Manager download Links for IBM tenants.
- Fixed an issue where large emails would fail to send due to RabbitMQ maximum message size.
- Fixed an issue where user personally identifiable information PII removal would not erase the UserPrincipalName from the database.
- Fixed an issue where license expiration comparison was not considering local date formatting and displaying an activation error when parsing dates.
- Fixed an issue that caused large domain-synchronization messages to not send, creating partial result sets for directory synchronization.
- Fixed an issue where email would not send from a web node with the background worker disabled.
- Custom dictionaries now support two-character entries.
- Fixed an issue where all IP address restrictions would not display more than 30 entries.

Heartbeat, Distributed Engines, and RPC

- Fixed an issue with high command throughput against SAP systems causing issues with logging those commands. We throttled the SAP password changers send-commands rate, which is configurable at Admin > Remote Password Changing > Configure Password Changing > SAP Account (or SAP SNC Account) > Advanced Settings. This delay defaults to 500 ms.
- Optimized the distributed engine administration page.
- Fixed an issue where distributed engines would fail to connect to the Azure service bus after updating.
- Fixed an issue where disabling RPC globally would also disable heartbeat globally for distributed engine sites.
- Fixed an issue where the distributed engine setup assistant page would show, even after a site connector had been created.
- Fixed an issue where successful password changes with failed dependencies would show as canceled.
- Fixed an issue where automatic user management may not be processed on an engine site.
- Fixed an issue where heartbeat would fail if a valid client certificate existed for the LDAP connection used to heartbeat. Heartbeat now attempts Kerberos first.

Installation, Upgrade, and Uninstall

None

Integration

- Fixed an issue where a CredSSP generic error returned when a ticketing system integration script failed. This now returns the specific error.
- Fixed an issue where using HSM integration without having performed a master encryption key rotation would cause performance issues.
- Resolved an issue where Connection Manager would not correctly present fields requiring a user prompt.

- Fixed an issue with misleading settings for ticket system configuration. It now always prompts for a site.
- Fixed an issue where "RADIUS default username" could not be set on existing users when enabling RADIUS.

Launchers and Protocol Handlers

- Fixed an issue where mapping a launcher port field failed with the error "this field must be a number."
- Fixed an issue where launcher restrictions were not letting users limit user input in some scenarios.
- Fixed an issue where using "run process as secret credentials" launcher option and enabling session recording could result in launcher startup failure due to an access denied error.
- Fixed an issue where session connector RDP launchers would fail if there was no domain on the secret.
- Fixed an issue that caused proxied launchers to not display the port field if the launcher was modified after creation.
- Fixed an issue where protocol handler would downgrade if a newer version than the version available from the server is used.
- Fixed an issue where the 64-bit installer of protocol handler would include a 32-bit executable. This could start a race condition, causing launchers to close.
- Fixed an issue where the session-connector protocol handler could update to the normal client version, disabling some session recording features.
- (GA only) Fixed an issue where some configurations of "hide launcher password" and "field view permissions" would evaluate incorrectly.

Networking

- Fixed an issue where checking for WinRM service status required administrative privileges.

Remote Access and Proxies

- Fixed an issue where SSH terminal froze when launching secrets with failing heartbeats.
- Fixed an issue where RDP proxy endpoints would respond as available even with an invalid configuration, causing proxy connection failures. Endpoints no longer respond as available if the configuration is invalid.
- Fixed an issue where SSH and RDP proxy connections failed due to Unicode characters in connection credentials.
- Fixed an issue where the jumpbox route connection forwarding could not forward across different network segments.
- Fixed an issue with SSH proxy performance related to the "hide passwords from SSH keystroke capture" setting.
- Fixed an issue where master encryption key rotation would fail if RDP proxy credentials were created but not used.
- Fixed an issue where the "hide Password from SSH keystroke capture" setting was always enabled.
- Fixed an issue where using GUI applications in an SSH session would not work with restricted command lists enabled.

- Fixed an issue where pressing the down arrow key in an SSH proxy session could crash the SSH proxy.
- Fixed an issue where SSH key authentication to SSH terminal would display incorrect prompts.
- Fixed an issue where the jumpbox route port range setting was not used correctly by engine proxies.
- (GA only) Fixed an issue where SSH key rotation would fail on the first attempt.

Reports

- Fixed an issue where the report "What folder permissions exist?" would not run on newer SQL Server versions.
- Fixed an issue with reports failing with a "not valid for reporting" error with older versions of SQL Server.
- Fixed issue causing the "reports" option to be incorrectly hidden.
- Fixed an issue with reports failing to send to email when no start date is defined but the stacked-column chart type is used.
- Fixed an issue where the "secrets with failed password change" report would show an incorrect value for DateRecorded.
- (GA only) Fixed an issue where reports could not be selected in the left navigation on some pages.

Secret Server Cloud

- Fixed a Secret Server Cloud issue where engines were incorrectly set to enable FIPS compliance, causing SSH cipher issues.

Secrets, Policies, and Templates

- Fixed an issue where secret erase would not run on a secret that had a previously changed password.
- Fixed an issue where user could not add a list field if the other non-list secret fields were not correctly configured.
- Fixed an issue where editing a secret template launcher mapping would remove the port. To restore the port to affected launchers, edit the mapping and re-save it.
- Fixed an issue where password fields would display as empty on the duplicate secret dialog box.
- Adjusted timeout behavior with bulk operations to provide more accurate statuses.
- Fixed an issue with exporting secret templates failing under some conditions.
- Fixed an issue where long URLs would affect page layout in the secret details view.

Session Recording

- Fixed an issue with advanced session recording agent installs always showing as a new entry. The entry now persists between upgrades.

Users and Groups

- Fixed an issue where column headings would disappear on the group membership assignment page.

UX/UI

None

Web Password Filler

- Fixed an issue where web password filler would default to the username field, regardless of the field mapping template settings.

Future and Recent Deprecations

Note: This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server: 11.2.000000 Release Notes (EA)

Release dates:

Early Availability May 5, 2022 (On-Premises)

Please see the "Secret Server: 11.2.000002 Release Notes (GA)" on page 1723 for EA too. On that page, any bug fixes that only apply to GA are marked as such.

Secret Server: 11.1.000012 Release Notes

Release date:

May 13, 2022 (On-Premises)

Important: We recommend all affected Secret Server on-premise customers upgrade immediately to version 11.1.000012.

The 11.1.000012 update resolves a **security vulnerability** that was discovered internally. The Common Vulnerability Scoring System (CVSS) rates the issue high (7.6). It impacts Secret Server On-Premises up to 11.1.000011.

This vulnerability has been patched in Secret Server Cloud, so there is no additional update to address it.

This vulnerability is patched in the version 11.2.000000 Early Adopter release.

Note: This release also includes some high priority bug fixes that have been through our full QA process.

Bugs

- Added a DataDeliveryTolerance application setting for distributed engines for troubleshooting proxy keystroke-recording issues.
- Fixed an issue where column headings would disappear on the group membership assignment page.
- Fixed an issue with reports failing with a "not valid for reporting" error with older versions of SQL Server.
- Fixed an issue where master encryption key rotation would fail if RDP proxy credentials were created but not used.
- Fixed an issue where "hide password from SSH keystroke capture" was always enabled.

- Fixed an issue where the login page would load slowly when using HSM integration.
- Fixed an issue where extend checkout would fail for users with view permission.
- .Fixed an issue where mapping a launcher port field would fail with the error "This field must be a number."
- Fixed a bug where updating node records during an upgrade upon application start or restart would log a "value cannot be null" error logs.
- Fixed an issue where launcher restrictions were not letting users restrict user input in some scenarios.
- Fixed an issue with SSH proxy performance related to the "hide Passwords from SSH keystroke capture" setting.
- Fixed an issue where the jumpbox route connection forwarding could not forward across different network segments.
- Fixed an issue that caused proxied launchers to stop working if modified after creation. A port field in the template was not visible in UI.
- Fixed an issue where attempting to cancel a character set validation rule and then saving resulted in an error.
- Fixed an issue where attempting to save a secret after the require comment timeout had expired would cause the interface to hang.
- Fixed an issue where the port would be removed from a secret template launcher mapping when edited. To restore the port to affected launchers, users had to edit the mapping and re-save it.
- Fixed an issue where applying a policy with the "web Launcher requires Incognito mode" setting during secret creation using the API would cause the creation to fail.
- Made changes to stop SSH terminal from freezing when a secret heartbeat is failing.
- Fixed an issue where an empty "items" property existed on api/v1/lists/{categorizedListid}.
- Fixed an issue where a CredSSP generic error would be returned when a ticketing system integration script failed. This now returns a more specific error.
- Fixed an issue where an incorrect identifying IP address was sent to Duo. This now sends the client address.
- Fixed an issue a single malfunctioning site would cause messages to stop processing for multiple sites.
- Fixed an issue where using HSM integration without having performed a master encryption key rotation would cause slow responsiveness.
- To address performance issues, we reduced the number of folders displayed in a folder tree when there are more than 1000 items.
- Updated the time zone database library with the latest time zone information.
- Optimized the distributed engine administration page.

Secret Server: 11.1.000007 Release Notes

> 11.1.000008 is a retroactive patch for this release. Please see "Secret Server 11.7.000001 Release Notes" on page 1596 for details.

Release dates:

January 25, 2022 (On-Premises)

January 15 2022 (Cloud) (unchanged from 11.1.000006)

Important: The 11.1.000007 update resolves a security vulnerability that was discovered during third-party penetration testing. The Common Vulnerability Scoring System (CVSS) rates the issue high (7.4). It impacts Secret Server On-Premises up to 11.1.000006. We recommend all affected Secret Server on-premise customers upgrade immediately to version 11.1.000007. This vulnerability does not apply to Secret Server Cloud, so there is not an update to address it.

Important: If you installed Secret Server as your default or top-level website and you have Privilege Manager (PM) and Secret Server installed together, you may experience the following issues after upgrading to .NET Framework 4.8:

- PM agents will not register.
- When a PM agent updates itself (using the agent utility), it states that there are zero policies to download.

If you believe this scenario applies to you, please contact Delinea Support **before** performing a .NET, Secret Server, or PM upgrade.

New Features and Enhancements

Master Encryption Key Rotation

Secret Server can now generate new master encryption keys via a rotation process. Previously, the master encryption key was generated at installation and rotation was not possible.

This feature is at Admin > Configuration on the Security tab in the Master Encryption Key Rotation section.

Note: Secret encryption uses a different key, and the independent secret key rotation is still available.

SSH Jumpbox Route Support

SSH jumpbox routes allow SSH terminal and proxy to connect through one or more jumpbox servers in line to a final connection when launching from that target server's secret. An *SSH jumpbox*, a type of *bastion host*, is a regular Linux server, accessible from the Internet, that is a gateway to other Linux machines on a private network using the SSH protocol.

Note: *Bastion* means *a projecting part of a fortification*. Bastion hosts are hardened and monitored servers that reside outside of an organization's security zone, usually exposed to the internet. SSH jumpboxes are also called *bastion hosts*, *jump hosts*, *jump boxes*, or *jump servers*. All jumpboxes are bastion hosts, but all bastion hosts are not necessarily jumpboxes.

RDP Clipboard and Drive Mapping

Gives the owner of a secret control of the RDP clipboard and drive mapping settings, restricting users from using their alternative settings. This provides administrators enforcement control over secrets and users' RDP launcher settings.

Checkout Time Indicator

There is now an indicator within secret details that shows the remaining time on a checkout and can extend the checkout if required. The indicator is color coded and counts down in days, hours, or minutes.

Click the timer for an option to extend the checkout if this is enabled. The setting to enable checkout extension is at Admin > Configuration on the General tab. Enable the "Enable Secret Check Out Extension" check box after clicking the Edit button.

Enhanced Diagnostic and Logging Functionality

The logging level of Secret Server web nodes and distributed engines are now centrally configurable and collectable. This feature is especially useful for large systems with many nodes and engines.

Configuration for the web nodes is found on the Server Nodes configuration page, alongside role settings. Configuration for distributed engines is found in the Distributed Engine configuration page. Log levels include: All, Debug, Info, Warn, Error, Off, and Not Set (the default). Previously, manual configuration file changes were required. “Not Set” relies on the configuration files for the logging level, which was the previous default behavior.

The diagnostic feature for collecting logs is improved and now gathers logs from all nodes and engines. This feature is at Admin > Diagnostics.

API Automatic Checkout

There are now automatic check in and check out parameters for secret API calls that can check in and out, leave comments, and force check in. The parameters are:

- `autocheckout=true` checks out the secret before performing any operations and then checks the secret back in afterwards.
- `forcecheckin=true` checks in the secret if it is currently checked out before performing any operations.
- The combination of `autocheckout=true` and `autocheckin=false` leaves the secret checked out after the operation completes.
- The combination of `autocheckout=true` and `autocheckin= true` checks out the secret before performing any operations and then checks the secret back in afterwards.
- `autocomment=[comment]` adds a check out or in comment. The comment must be in a URL-escaped format. For instance spaces are changed to `%20`.

This functionality is currently available on the following endpoints:

- `api/v1/secrets/{id}/`
- `api/v1/secrets/{id}/change-password/`
- `api/v1/secrets/{id}/email`
- `api/v1/secrets/{id}/expiration`
- `api/v1/secrets/{id}/expire/`
- `api/v1/secrets/{id}/fields/{slug}/`
- `api/v1/secrets/{id}/fields/{slug}/`
- `api/v1/secrets/{id}/fields/{slug}/list`
- `api/v1/secrets/{id}/fields/{slug}/listdetails`
- `api/v1/secrets/{id}/general`

- `api/v1/secrets/{id}/heartbeat/`
- `api/v1/secrets/{id}/restricted/`
- `api/v1/secrets/{id}/restricted/fields/{slug}/`
- `api/v1/secrets/{id}/rpc-script-secrets`
- `api/v1/secrets/{id}/security-general`
- `api/v1/secrets/{id}/settings`
- `api/v1/secrets/{id}/ssh-restricted-commands`
- `api/v1/secrets/{id}/state`
- `api/v1/secrets/{id}/summary/`
- `api/v1/secrets/{secretId}/update-ssh-restricted-commands`
- `api/v1/secrets/rdpproxy`
- `api/v1/secrets/sshproxy`
- `api/v1/secrets/sshterminal`
- `api/v2/secrets/{id}/email`
- `api/v2/secrets/{id}/general`
- `api/v2/secrets/{id}/rpc-script-secrets`
- `api/v2/secrets/{id}/security-general`
- `api/v3/secrets/{id}/security-approval`
- `api/v3/secrets/{id}/security-checkout`

General

- Added support for RabbitMQ version 3.9.5 with RabbitMQ Helper.
- Updated syslog over TCP to meet the requirements of RFC-6587. Added a newline terminator at the end of each syslog message per RFC 6587. You can disable this by setting the "System Log force line feed end of line" advanced configuration setting to "False."
- Upgraded RADIUS authentication to include EAP-TTLS-PAP.
- Added options to prevent login via OAuth if SAML is enabled.
- Improved audit logs with addition of username and display name data.
- Added token (such as `$EventUsername`) translation to the input value for the secret field filter in event pipeline policies.
- The "Restrict by" setting in launchers and now use notes, text fields, or list fields.

UI/UX

- Added accessibility controls across Secret Server.
- Added a "quick launcher" feature to the secret table view, providing faster access to the launchers from search results or folders.
- Added screen reader hints.
- Improved wording in the UI description for managed directory services groups.
- UI description for user preference email settings now says users with view permissions see secret notifications.
- Improved the UI for assigning multiple groups to a user from the user management page.
- Improved the UI adding permissions to folders.
- Improved the secret template administration pages.
- Added a copy password button to the secret password history (Secret Item Value History popup). The history now has a multi-line text box, making it easier to manipulate text strings.
- Added a `#TIMEZONE` parameter to the custom report builder. This inserts the user time zone (formatted for SQL) into the report. This can be used with SQL syntax for converting dates, such as `CONVERT(DATE, DateRecorded AT TIME ZONE ''UTC'' AT TIME ZONE #TIMEZONE) as DateRecorded`
- Added a folder search to the folder picker.

API

- The `GET /folders` API endpoint can now optionally return only root folders by using the new query parameter `filter.OnlyIncludeRootFolders=true`.

Bug Fixes

Access Requests and Secret Workflows

- Fixed an issue where comments for checked-out secrets appear in the preview panel.
- Fixed an issue when including owners as reviewers was only updated if another part of the step was updated.
- Fixed the issue where the max allowed approvers setting was being reset to 1 instead of the number of approvers.

API

- Fixed an issue with API secret endpoints not returning a 400 bad request statement when an approval workflow is used.

Alerts, Auditing, and Logs

- Fixed an issue where auto export would report an error in the system log when it is not enabled.
- Fixed an issue with the historical data import for Privileged Behavior Analytics not working.
- Fixed an issue where inbox templates would not load if they were in a language different to the application.

- Fixed a performance issue when retrieving secret audits with large numbers of audits and secret sessions associated with the secret.
- Fixed an issue where AccessConfigs is null for some customers. The fix checks before using the values in AccessConfigs and creating additional logs.

Authentication, Login, and Directory Services

- Fixed an issue where OAuth tokens generated with a maximum lifetime would sometimes return an error.
- Increased the size limit of SAML certificate storage to accommodate larger certificates.
- Fixed an issue where certain LDAP directories could not synchronize because they did not support the SearchOptions control.
- Fixed an issue with bulk operations to disable two-factor authentication not running on the User Administration page.
- Fixed an Secret Server Cloud error when users attempted to use the "lost my phone" feature.
- Improved the AD sync error messaging by separating users from domains that failed to sync from those whose domains have no users in the latest sync, delivering separate error messages for each.
- Fixed a login issue by enforcing an existing restriction of application accounts from logging into the UI for SAML authentication.
- Fixed an issue with AD and LDAP Secret Server users not being able to successfully authenticate with a public key over SSH Terminal.
- Fixed an issue when 2FA was enabled for users with no entry in the tbuser fields causing login failure.
- Fixed an issue where duplicate item rows were added when generating a SSH key and passing to the passphrase. The duplicate caused an error in validation that expected only one row per field.
- Fixed an issue with AD sync over a distributed engine with "TLS error auditing" enabled that caused "disable user management" to stop working.

Discovery

- Fixed an issue with discovery host-range mapping validation.
- Fixed an issue where a mouseover tool tip was not showing the full OU path on the Discovery Domain Scope page for excluded OUs.
- Fixed an issue where previously scanned Linux machines were not displayed in discovery network view.
- Fixed an issue with discovery import rules for windows local account templates that prevented successful account import.
- Fixed an issue where an invalid secret template configuration caused a page load error when navigating to the Discovery Network View page.

Encryption, Passwords, and Certificates

- Fixed an issue with the password report in the dashboard UI to properly update the date.
- Fixed an issue where the "Prevent Username in Password" setting was not working for a password template.

Event Subscriptions and Pipelines

- Reduced the EventQueue "maximum batches per job" default from unlimited to 100. This is an advanced configuration item.
- Fixed an issue that prevented recording an audit log and sending an event subscription when "Require Two Factor for these Login Types" is set to "Web Services Login Only."

Folders

Fixed an issue where folder owners cannot move secrets to a folder if the owner does not have access to the folder's parent folder.

General

- Added table monitoring for DIM for new tables.
- Fixed an incorrect error code displayed when navigating to a URL containing invalid characters. Navigating to the URL now displays a 404 error.
- Fixed an issue with unclear licensing errors. Highlighted the error and clarified working.
- Fixed an issue with non-English languages displaying the product name incorrectly.
- Fixed a CVSS 7.4 security issue. It impacts Secret Server On-Premises up to 11.1.000006. See the important note at the top of these release notes.

Heartbeat, Distributed Engines, and RPC

- Fixed an issue where changes to proxy endpoint settings would not be audited on distributed engine proxy endpoints.
- Fixed an issue with distributed engine activation requiring MSDTC to be enabled.
- Fixed an issue with dropped connections to RabbitMQ causing worker process connections to RabbitMQ to accumulate additional connection channels, impacting connectivity.
- Fixed an issue where changing a dependency group site assignment to "Use Site from Secret" would throw an application error.
- Fixed an issue where a secret would be queued for password changing every password changing interval due to an invalid change schedule setting.
- Fixed an issue on the Secret Remote Password Changing settings tab where the option to remove associated secrets would not display.
- Fixed an issue with the Test Dependency button on the Secret Dependency view did not work correctly on the New UI.
- Fixed an issue where bulk operation "Assign to Site" required heartbeat or remote password changing to be enabled on the template
- Fixed an issue on the RPC configuration auto-change schedule not honoring a scheduled "when password expires" change.

- Fixed an issue that occurred when running RPC on a secret and then switching to view another secret–if the first secret's RPC fails, the information is displayed on the second secret.
- Fixed an issue where RPCNextAttemptTime does not update when adjusting the auto change schedule after a failed RPC attempt.
- Fixed an issue in discovery where the Secret Type field does not provide dropdown options in the UI if RPC is disabled on the secret template's password changer.
- Fixed an issue with the application path of Secret Server being used in a distributed engine for ComPlus dependencies.

Installation, Upgrade, and Uninstall

None

Launchers

- Fixed an issue where the "Hide Launcher Password" setting would hide the password field from a secret owner when the field also has "View Requires Edit" enabled.
- Fixed an error when a machine list restricts an input field on a secret launcher.
- Fixed an issue with protocol handler that caused certificate errors from the IDP or delays in validating the CRL.

Remote Access and Proxies

- Fixed an issue with proxied SSH connections to slow devices where "Connect As" commands were not inputted correctly.
- Fixed an issue where some key-exchange algorithms were not supported for various SSH tasks, including proxy, discovery, heartbeat and password changing.
- Fixed an issue where SSH command restrictions using allowed command lists would close the connection when trying to navigate the command restrictions menus.
- Fixed an issue with automatic sudo elevation during SSH proxy sessions identifying password prompts incorrectly and attempting to enter the password.
- Fixed an issue with SSH proxy session performance when sending or receiving large quantities of data.
- Fixed an issue where using SSH Blocked Command lists prevented exiting text editors in an SSH session.
- Fixed an issue with reports using date range filters. The local timezone is now correctly filters the report, as opposed to UTC.
- Fixed a memory leak issue in SSH proxy on Web node connections.
- Fixed an issue where PuTTY did not load the default session logging location.
- Fixed an issue where SSH Terminal was locked out due to an authorization issue when launching a secret.

Reports

- Fixed an issue with users not receiving report emails.
- Fixed an issue where scheduled reports failed to send emails when the report file size was too large, producing an error.

Secret Server Cloud

Fixed an issue where secret key rotation could trigger database growth when used with KMS key protection.

Secrets, Policies, and Templates

- Fixed an issue with the secret template interface that prevented users from setting a default deprived secret. This was possible in the Classic UI.
- Clarified an error when changing the field type on a secret template that occurs if the field is mapped to the "Restrict By" setting of a launcher.
- Fixed an issue on the Secret General tab where the icon indicating that a session is not recorded did not display on Web Password Filler launchers.
- Fixed an issue with secret search where results appeared to be matching any word in the search text instead of all words.
- Fixed an issue with secret import not using the site from the import XML unless the site was also added during the same import. Import will now use the site ID on the secret if sites were not also imported.
- Fixed an issue where a checked-in secret's password is visible. Secret Server now displays an error if the user tries to click the view password link on a the secret.
- Fixed an issue that prevented access to a secret in "unlimited admin mode" if the secret is checked-out and requires a comment.
- Fixed an issue with secret search when searching an encrypted secret field with partial matches of some terms.
- Fixed an issue with the Secret Settings interface showing incorrect settings when a secret policy was enforced.

Session Recording

None

Users and Groups

- Fixed an issue with automatic user management not correctly enabling or re-enabling users in some configurations during SAML login.
- Fixed an issue in the UI where groups with long names obscured permission dropdown-list options. Long group names now text wrap to allow viewing.

UX/UI

Fixed an issue where switching languages would display the incorrect OEM logo and product name in the classic UI.

Web Password Filler

None

Future and Recent Deprecations

Note: This section describes planned future deprecation of feature or platform support in Secret Server.

- Internet Explorer 11. Support for Internet Explorer 11 ends on 31 August 2021. Secret Server releases after that date will not support Internet Explorer 11.
- Secret Server Classic UI. The Classic UI option in Secret Server is scheduled to be removed in Q2 2022. After that time, the New UI will be the only available UI option in Secret Server.

Secret Server: 11.1.000006 Release Notes

Release dates:

January 11, 2022 (On-Premises)

January 15 2022 (Cloud)

Important: If you installed Secret Server as your default or top-level website and you have Privilege Manager (PM) and Secret Server installed together, you may experience the following issues after upgrading to .NET Framework 4.8:

- PM agents will not register.
- When a PM agent updates itself (using the agent utility), it states that there are zero policies to download.

If you believe this scenario applies to you, please contact Delinea Support **before** performing a .NET, Secret Server, or PM upgrade.

New Features and Enhancements

Master Encryption Key Rotation

Secret Server can now generate new master encryption keys via a rotation process. Previously, the master encryption key was generated at installation and rotation was not possible.

This feature is at Admin > Configuration on the Security tab in the Master Encryption Key Rotation section.

Note: Secret encryption uses a different key, and the independent secret key rotation is still available.

SSH Jumpbox Route Support

SSH jumpbox routes allow SSH terminal and proxy to connect through one or more jumpbox servers in line to a final connection when launching from that target server's secret. An *SSH jumpbox*, a type of *bastion host*, is a regular Linux server, accessible from the Internet, that is a gateway to other Linux machines on a private network using the SSH protocol.

Note: *Bastion* means *a projecting part of a fortification*. Bastion hosts are hardened and monitored servers that reside outside of an organization's security zone, usually exposed to the internet. SSH jumpboxes are also called *bastion hosts*, *jump hosts*, *jump boxes*, or *jump servers*. All jumpboxes are bastion hosts, but all bastion hosts are not necessarily jumpboxes.

RDP Clipboard and Drive Mapping

Gives the owner of a secret control of the RDP clipboard and drive mapping settings, restricting users from using their alternative settings. This provides administrators enforcement control over secrets and users' RDP launcher settings.

Checkout Time Indicator

There is now an indicator within secret details that shows the remaining time on a checkout and can extend the checkout if required. The indicator is color coded and counts down in days, hours, or minutes.

Click the timer for an option to extend the checkout if this is enabled. The setting to enable checkout extension is at Admin > Configuration on the General tab. Enable the "Enable Secret Check Out Extension" check box after clicking the Edit button.

Enhanced Diagnostic and Logging Functionality

The logging level of Secret Server web nodes and distributed engines are now centrally configurable and collectable. This feature is especially useful for large systems with many nodes and engines.

Configuration for the web nodes is found on the Server Nodes configuration page, alongside role settings. Configuration for distributed engines is found in the Distributed Engine configuration page. Log levels include: All, Debug, Info, Warn, Error, Off, and Not Set (the default). Previously, manual configuration file changes were required. "Not Set" relies on the configuration files for the logging level, which was the previous default behavior.

The diagnostic feature for collecting logs is improved and now gathers logs from all nodes and engines. This feature is at Admin > Diagnostics.

API Automatic Checkout

There are now automatic check in and check out parameters for secret API calls that can check in and out, leave comments, and force check in. The parameters are:

- `autocheckout=true` checks out the secret before performing any operations and then checks the secret back in afterwards.
- `forcecheckin=true` checks in the secret if it is currently checked out before performing any operations.
- The combination of `autocheckout=true` and `autocheckin=false` leaves the secret checked out after the operation completes.
- The combination of `autocheckout=true` and `autocheckin= true` checks out the secret before performing any operations and then checks the secret back in afterwards.
- `autocomment=[comment]` adds a check out or in comment. The comment must be in a URL-escaped format. For instance spaces are changed to `%20`.

This functionality is currently available on the following endpoints:

- api/v1/secrets/{id}/
- api/v1/secrets/{id}/change-password/
- api/v1/secrets/{id}/email
- api/v1/secrets/{id}/expiration
- api/v1/secrets/{id}/expire/
- api/v1/secrets/{id}/fields/{slug}/
- api/v1/secrets/{id}/fields/{slug}/
- api/v1/secrets/{id}/fields/{slug}/list
- api/v1/secrets/{id}/fields/{slug}/listdetails
- api/v1/secrets/{id}/general
- api/v1/secrets/{id}/heartbeat/
- api/v1/secrets/{id}/restricted/
- api/v1/secrets/{id}/restricted/fields/{slug}/
- api/v1/secrets/{id}/rpc-script-secrets
- api/v1/secrets/{id}/security-general
- api/v1/secrets/{id}/settings
- api/v1/secrets/{id}/ssh-restricted-commands
- api/v1/secrets/{id}/state
- api/v1/secrets/{id}/summary/
- api/v1/secrets/{secretId}/update-ssh-restricted-commands
- api/v1/secrets/rdpproxy
- api/v1/secrets/sshproxy
- api/v1/secrets/sshterminal
- api/v2/secrets/{id}/email
- api/v2/secrets/{id}/general
- api/v2/secrets/{id}/rpc-script-secrets
- api/v2/secrets/{id}/security-general
- api/v3/secrets/{id}/security-approval
- api/v3/secrets/{id}/security-checkout

General

- Added support for RabbitMQ version 3.9.5 with RabbitMQ Helper.

- Updated syslog over TCP to meet the requirements of RFC-6587. Added a newline terminator at the end of each syslog message per RFC 6587. You can disable this by setting the "System Log force line feed end of line" advanced configuration setting to "False."
- Upgraded RADIUS authentication to include EAP-TTLS-PAP.
- Added options to prevent login via OAuth if SAML is enabled.
- Improved audit logs with addition of username and display name data.
- Added token (such as `$EventUsername`) translation to the input value for the secret field filter in event pipeline policies.
- The "Restrict by" setting in launchers and now use notes, text fields, or list fields.

UI/UX

- Added accessibility controls across Secret Server.
- Added a "quick launcher" feature to the secret table view, providing faster access to the launchers from search results or folders.
- Added screen reader hints.
- Improved wording in the UI description for managed directory services groups.
- UI description for user preference email settings now says users with view permissions see secret notifications.
- Improved the UI for assigning multiple groups to a user from the user management page.
- Improved the UI adding permissions to folders.
- Improved the secret template administration pages.
- Added a copy password button to the secret password history (Secret Item Value History popup). The history now has a multi-line text box, making it easier to manipulate text strings.
- Added a `#TIMEZONE` parameter to the custom report builder. This inserts the user time zone (formatted for SQL) into the report. This can be used with SQL syntax for converting dates, such as `CONVERT(DATE, DateRecorded AT TIME ZONE ''UTC'' AT TIME ZONE #TIMEZONE) as DateRecorded`
- Added a folder search to the folder picker.

API

- The `GET /folders` API endpoint can now optionally return only root folders by using the new query parameter `filter.OnlyIncludeRootFolders=true`.

Bug Fixes

Access Requests and Secret Workflows

- Fixed an issue where comments for checked-out secrets appear in the preview panel.
- Fixed an issue when including owners as reviewers was only updated if another part of the step was updated.
- Fixed the issue where the max allowed approvers setting was being reset to 1 instead of the number of approvers.

API

- Fixed an issue with API secret endpoints not returning a 400 bad request statement when an approval workflow is used.
- Fixed an issue with the SOAP API Method `AddNewSecret` that threw an object reference error when setting `SecretSettings.IsChangeToSettings = $true`.

Alerts, Auditing, and Logs

- Fixed an issue where auto export would report an error in the system log when it is not enabled.
- Fixed an issue with the historical data import for Privileged Behavior Analytics not working.
- Fixed an issue where inbox templates would not load if they were in a language different to the application.
- Fixed a performance issue when retrieving secret audits with large numbers of audits and secret sessions associated with the secret.
- Fixed an issue where AccessConfigs is null for some customers. The fix checks before using the values in AccessConfigs and creating additional logs.

Authentication, Login, and Directory Services

- Fixed an issue where OAuth tokens generated with a maximum lifetime would sometimes return an error.
- Increased the size limit of SAML certificate storage to accommodate larger certificates.
- Fixed an issue where certain LDAP directories could not synchronize because they did not support the SearchOptions control.
- Fixed an issue with bulk operations to disable two-factor authentication not running on the User Administration page.
- Fixed an Secret Server Cloud error when users attempted to use the "lost my phone" feature.
- Improved the AD sync error messaging by separating users from domains that failed to sync from those whose domains have no users in the latest sync, delivering separate error messages for each.
- Fixed a login issue by enforcing an existing restriction of application accounts from logging into the UI for SAML authentication.
- Fixed an issue with AD and LDAP Secret Server users not being able to successfully authenticate with a public key over SSH Terminal.
- Fixed an issue when 2FA was enabled for users with no entry in the tbuser fields causing login failure.
- Fixed an issue where duplicate item rows were added when generating a SSH key and passing to the passphrase. The duplicate caused an error in validation that expected only one row per field.
- Fixed an issue with AD sync over a distributed engine with "TLS error auditing" enabled that caused "disable user management" to stop working.

Discovery

- Fixed an issue with discovery host-range mapping validation.
- Fixed an issue where a mouseover tool tip was not showing the full OU path on the Discovery Domain Scope page for excluded OUs.
- Fixed an issue where previously scanned Linux machines were not displayed in discovery network view.
- Fixed an issue with discovery import rules for windows local account templates that prevented successful account import.
- Fixed an issue where an invalid secret template configuration caused a page load error when navigating to the Discovery Network View page.

Encryption, Passwords, and Certificates

- Fixed an issue with the password report in the dashboard UI to properly update the date.
- Fixed an issue where the "Prevent Username in Password" setting was not working for a password template.

Event Subscriptions and Pipelines

- Reduced the EventQueue "maximum batches per job" default from unlimited to 100. This is an advanced configuration item.
- Fixed an issue that prevented recording an audit log and sending an event subscription when "Require Two Factor for these Login Types" is set to "Web Services Login Only."

Folders

Fixed an issue where folder owners cannot move secrets to a folder if the owner does not have access to the folder's parent folder.

General

- Added table monitoring for DIM for new tables.
- Fixed an incorrect error code displayed when navigating to a URL containing invalid characters. Navigating to the URL now displays a 404 error.
- Fixed an issue with unclear licensing errors. Highlighted the error and clarified working.

Heartbeat, Distributed Engines, and RPC

- Fixed an issue where changes to proxy endpoint settings would not be audited on distributed engine proxy endpoints.
- Fixed an issue with distributed engine activation requiring MSDTC to be enabled.
- Fixed an issue with dropped connections to RabbitMQ causing worker process connections to RabbitMQ to accumulate additional connection channels, impacting connectivity.
- Fixed an issue where changing a dependency group site assignment to "Use Site from Secret" would throw an application error.

- Fixed an issue where a secret would be queued for password changing every password changing interval due to an invalid change schedule setting.
- Fixed an issue on the Secret Remote Password Changing settings tab where the option to remove associated secrets would not display.
- Fixed an issue with the Test Dependency button on the Secret Dependency view did not work correctly on the New UI.
- Fixed an issue where bulk operation "Assign to Site" required heartbeat or remote password changing to be enabled on the template
- Fixed an issue on the RPC configuration auto-change schedule not honoring a scheduled "when password expires" change.
- Fixed an issue that occurred when running RPC on a secret and then switching to view another secret—if the first secret's RPC fails, the information is displayed on the second secret.
- Fixed an issue where RPCNextAttemptTime does not update when adjusting the auto change schedule after a failed RPC attempt.
- Fixed an issue in discovery where the Secret Type field does not provide dropdown options in the UI if RPC is disabled on the secret template's password changer.
- Fixed an issue with the application path of Secret Server being used in a distributed engine for ComPlus dependencies.

Installation, Upgrade, and Uninstall

None

Launchers

- Fixed an issue where the "Hide Launcher Password" setting would hide the password field from a secret owner when the field also has "View Requires Edit" enabled.
- Fixed an error when a machine list restricts an input field on a secret launcher.
- Fixed an issue with protocol handler that caused certificate errors from the IDP or delays in validating the CRL.

Remote Access and Proxies

- Fixed an issue with proxied SSH connections to slow devices where "Connect As" commands were not inputted correctly.
- Fixed an issue where some key-exchange algorithms were not supported for various SSH tasks, including proxy, discovery, heartbeat and password changing.
- Fixed an issue where SSH command restrictions using allowed command lists would close the connection when trying to navigate the command restrictions menus.
- Fixed an issue with automatic sudo elevation during SSH proxy sessions identifying password prompts incorrectly and attempting to enter the password.
- Fixed an issue with SSH proxy session performance when sending or receiving large quantities of data.
- Fixed an issue where using SSH Blocked Command lists prevented exiting text editors in an SSH session.

- Fixed an issue with reports using date range filters. The local timezone is now correctly filters the report, as opposed to UTC.
- Fixed a memory leak issue in SSH proxy on Web node connections.
- Fixed an issue where PuTTY did not load the default session logging location.
- Fixed an issue where SSH Terminal was locked out due to an authorization issue when launching a secret.

Reports

- Fixed an issue with users not receiving report emails.
- Fixed an issue where scheduled reports failed to send emails when the report file size was too large, producing an error.

Secret Server Cloud

Fixed an issue where secret key rotation could trigger database growth when used with KMS key protection.

Secrets, Policies, and Templates

- Fixed an issue with the secret template interface that prevented users from setting a default deprived secret. This was possible in the Classic UI.
- Clarified an error when changing the field type on a secret template that occurs if the field is mapped to the "Restrict By" setting of a launcher.
- Fixed an issue on the Secret General tab where the icon indicating that a session is not recorded did not display on Web Password Filler launchers.
- Fixed an issue with secret search where results appeared to be matching any word in the search text instead of all words.
- Fixed an issue with secret import not using the site from the import XML unless the site was also added during the same import. Import will now use the site ID on the secret if sites were not also imported.
- Fixed an issue where a checked-in secret's password is visible. Secret Server now displays an error if the user tries to click the view password link on a the secret.
- Fixed an issue that prevented access to a secret in "unlimited admin mode" if the secret is checked-out and requires a comment.
- Fixed an issue with secret search when searching an encrypted secret field with partial matches of some terms.

Session Recording

None

Users and Groups

- Fixed an issue with automatic user management not correctly enabling or re-enabling users in some configurations during SAML login.
- Fixed an issue in the UI where groups with long names obscured permission dropdown-list options. Long group names now text wrap to allow viewing.

UX/UI

Fixed an issue where switching languages would display the incorrect OEM logo and product name in the classic UI.

Web Password Filler

None

Future and Recent Deprecations

Note: This section describes planned future deprecation of feature or platform support in Secret Server.

- Internet Explorer 11. Support for Internet Explorer 11 ends on 31 August 2021. Secret Server releases after that date will not support Internet Explorer 11.
- Secret Server Classic UI. The Classic UI option in Secret Server is scheduled to be removed in Q2 2022. After that time, the New UI will be the only available UI option in Secret Server.

Secret Server: 11.0.000008 Release Notes

Release date:

- September 30, 2021 (on-premises and cloud)

Important: The 11.000008 update resolves a **serious security vulnerability**. Please carefully read the following *High Security Issue* section. **We strongly recommend all affected Secret Server customers upgrade at their earliest opportunity**.

Important: If you installed Secret Server as your default or top-level website and you have Privilege Manager (PM) and Secret Server installed together, you may experience the following issues after upgrading to .NET Framework 4.8:

- PM agents will not register.
- When a PM agent updates itself (using the agent utility), it states that there are zero policies to download.

If you believe this scenario applies to you, please contact Delinea Support **before** performing a .NET, Secret Server, or PM upgrade.

High Security Issue

Overview

The 11.000008 update resolves a high security vulnerability for customers using OpenLDAP directory services with Secret Server.

This issue is rated **High** with a 8.6 Common Vulnerability Scoring System (CVSS) score. Please see the CVSS Calculator for the issue.

Scope

This vulnerability applies when all of the following are true:

- The user is authenticated via an OpenLDAP domain.
- The user is accessing Secret Server via the API or a tool that uses the API, such as Connection Manager, Web Password Filler, or the Secret Server mobile app.
- The LDAP domain is configured to allow unauthenticated bindings. For example, the presence of `allow bind_anon_cred` in the OpenLDAP configuration.

Remediation

Installing the update resolves the issue. In the meantime, as a temporary workaround, we suggest disabling unauthenticated binding in your OpenLDAP directory configuration.

Known Issues

Note: All the original release notes for 11.0.000007 have been moved below for convenience and clarity.

Connection Manager Version 1.6.2 Required for Customers Using Secret Server 11.0

Details:

- Intermittent Issue: On a macOS, when run in protocol handler mode, copy and paste does not work. The app does not crash but the copy and paste action is not successful. If the app is “unlocked,” copy, cut, and paste work. For your Secret Server connection to “unlock” it, expand the left navigation panel, and double click on the “lock” icon.
- Intermittent Issue On a macOS, while connected to any Secret Server, if the user edits the same Secret Server name (via the Secret Server connections menu), the app could get stuck in a “connecting…” state.
- Intermittent Issue: On a macOS, some users may experience issues connecting to Secret Server via a Web login (SAML) where the connection stays in a “connecting…” state.
- Upgrade Issues: In some situations, the automatic upgrade may not run the installer after downloading the MSI or package files. On macOS, when upgrading from v1.6.0, the automatic update fails to launch the installer. On Windows, the auto update may not launch the installer. In that case, please download and run the installer manually or via a software deployment tool.

  Installer download links:

  - macOS
  - Windows

New Features and Enhancements

API

Secrets are now searchable by the full path. This removes the need to know the ID to locate the intended secret.

Encrypted Secret Export

Secret Server now allows you to schedule encrypted exports of secret data to external storage.

General

- Updated ESXi integration support via an update of PowerCLI 12.
- Updated PuTTY to version .074.
- Added SNC support for communication between SAP and Secret Server for heartbeat and remote password changing.
- Added support for metadata filters on event pipelines.
- Added the ability to intercept a sudo or su command in a proxied session and inject the session password directly from Secret Server, not disclosing the password to the user.
- Added support for displaying both username and display name in audit logs.
- Added SMB fallback support for local Windows account heartbeats.
- Added a discovery threshold configuration that controls when a dependency is deactivated if it is not found in a scan. The threshold is set to the times a dependency can be missing prior to deactivation. It can also be set to "never" to prevent deactivation.

Inbox

The inbox now provides a customizable toolset to manage how email and notifications are sent and received by users. Inbox allows for configuration of notification scheduling, collecting notifications into digests, creation of message templates and rules, and more.

Lists

Secrets now support configurable data lists. Users with the "apply lists to secrets via secret policies" permission can create a list. This provides an easy mechanism for secret owners to simply choose from provided lists, such as a list of machines the user can select. Additionally, lists can be used for allow and blocklists allowing for control over what the secret owner can access.

Proxy Generic Connections

Secret Server can now securely tunnel a connection to servers operating on a variety of protocols.

Secret Erase

Users with the permissions to use the secret erase feature can permanently erase data from a secret. This provides Secret Server a method to purge secret data without reconciliation of the erased data. The existing "delete" function (now called "deactivate") allows you to "undelete" (reactivate). Secret erase can be audited as an event.

SSH Key Discovery

Secret Server discovery can now discover SSH public keys by scanning key locations on Linux and Unix servers.

User Interface

- Added dependency status reports. The reports are:
  - Overview: Shows how many dependencies failed, succeeded, and were not run.
  - Status: Shows how many dependencies failed, succeeded, and were not run by clickable secret, secret dependency group, and site.
  - Failed by secret: Shows doughnut graphic with secret and fail count.
  - Not run by secret: Shows the not run count by secret.
- Renamed the "Viewing Password Requires Edit" setting to "Hide Launcher Password" in the secret policies editor to improve clarity.

Bug Fixes

Note: The same line item may appear in more than one section when it applies to both.

Access Requests and Secret Workflows

- Fixed an issue where the "Revoke" button was being displayed in the UI after a request had already expired. The button should not be available to users past expiration.
- Fixed an issue where a "No Permission" page is displayed incorrectly when checking-in a secret with "change on check-in" enabled.

Alerts, Events, and Logging

- Fixed an issue where if the browser is set to a non-English language, a failed login attempt causes invalid system logging. The fix changes the character type from text to nvchar to produce successful logging entries. For this data type conversion, we recommend that customers with lots of data may want to run the SQL delta ahead of an upgrade to prevent potential SQL timeouts during upgrade.

API

- Fixed an issue where API was returning null for createDate on SecretSummary queries.
- Fixed an issue that resulted in receiving a 500 error when using the REST API to enable check-in on a secret using the PATCH method /secrets/{id}/security-checkout.
- Fixed an issue where the REST API was returning "Object reference not set to an instance of an object." against the /directory-services/domains/{domainid}/group.
- Fixed an issue for the API endpoint, workflow, or template throwing an exception when the take parameter was not specified.
- Fixed a POST issue in secret-templates REST endpoint that did not properly validate the editablePermission property on fields.
- Fixed an issue where a secret policy which enforces 'View Password Requires Edit' prevents API access to the view the password.

Authentication, Login, and Directory Services

- Fixed an issue with post authentication for SAML.
- Fixed an issue where if the browser is set to a non-English language, a failed login attempt causes invalid system logging. The fix changes the character type from text to nvchar to produce successful logging entries. For this data type conversion, we recommend that customers with lots of data may want to run the SQL delta ahead of an upgrade to prevent potential SQL timeouts during upgrade.
- Fixed an issue where when a distributed engine was used as a proxy server the SSH terminal gave an invalid error during authentication for public keys.
- Fixed an issue where occasionally users that synced via Active Directory were not getting assigned default role (s).
- Fixed an issue with Active Directory synchronization throwing a credential validation error when the sync credentials are from another domain after creating relationship between domains.
- Added a notice in the UI for directory services group sync search results being limited to 1000 groups.
- Fixed an issue where Active Directory group names were showing with their pre-Windows 2000 name instead of the group name.
- Fixed an issue where Secret Server Cloud users were presented with a misleading tooltip for user synchronization interval options in directory services configuration. An improved tooltip has been provided to inform the user the limitations of the configuration more accurately in Secret Server Cloud.

Discovery

- Fixed an issue in discovery when using LDAPS where discovery attempts to use port 389 despite the user having selected LDAPS port 636.
- Fixed an issue with saving IP addresses additions to the manual host range within the discovery scanner. A message is now provided to inform the user that the manual host ranges should be set on the discovery source scanner, rather than as a default setting on the scanner itself.
- Fixed an issue that could occasionally cause dependencies to get disabled by a computer discovery scan. The fix adds a discovery configuration for a threshold to improve control when a dependency is deactivated if not found in a scan. The threshold can be set to ‘Never’ to prevent disabling or a count for times a dependency is not found.
- Fixed a discovery issue that could cause background processes to stop.
- Fixed an issue where discovery was retaining too many historical records for each host. We added an advanced configuration setting to define the number of records.
- Fixed a discovery issue where when the discovery sourceid was equal to 9 or 10, after the domain scan completed to get machines, the Service Account tab in the UI would not load re-scan buttons.
- Fixed an issue where the schedule task scanner did not work for the domain discovery source with two UPNs.
- Fixed an issue with discovery for scheduled task scanning across multiple trusted domains.
- Fixed a discovery issue where similar OU names resulted in discovery importing accounts from an OU with a similar name to the targeted OU.

Event Subscriptions and Pipelines

- Fixed an issue that could be encountered with event subscriptions after converting a secret's template. An improved message is now provided in the UI when converting secret templates to inform the user what actions needs to take place to prevent the issue.
- Fixed an issue with event pipelines receiving an application error when trying to add tasks in a policy if the policy has a deactivated pipeline.
- Fixed an event subscriptions issue of not initiating for engine events.

Folders

- Fixed an issue when editing folder permissions and receiving no search results for groups that contained a '+' in the group name.
- Fixed an issue that prevented the ability to sort folders by alphabetical order on name in the Favorites tab.
- Fixed an issue where the folder name may not display in the secret grid.
- Fixed an issue where right clicking secret folders in the UI did not have the "View Details" option available for users with view permissions.
- Fixed an issue where editing folder sharing permissions removed restricted template options in the new UI.
- Fixed an issue that caused an application error when changing a folder's secret permissions to "none."

General

- Fixed an issue that caused a 500 error if certain password sequences were used when password validation settings were enabled to prevent password sequences.
- Fixed an issue that prevented content deletion in metadata fields.
- Fixed an issue where pressing enter on the keyboard when using the main application search box did not display results in the secret grid but only in the dropdown.
- Fixed an issue where the "Copy to Clipboard" button was not appearing on fields that contained a character count beyond the max field character display.
- Fixed an issue where the "Options" menu on pages is automatically hidden in the UI if no options are present to display to the user.
- Fixed an issue where sessions created through the Secret Server terminal "Launch" command were not getting returned by session search when filtering by secret name, secret field value, or user.
- Fixed an issue where on the home page engine status widget was not displaying the correct status in the "Connection Status 1" column. In the fix, this column has been renamed to 'Activation Status'.
- Fixed an invalid audit message entry when a user initiates "Change Password Now" using the "Randomly Generated" option.
- Fixed an issue where changing the name of a launcher did not update the name of the launcher on the secret.
- Fixed a mouseover label on "RADIUS User Name" on the Users page. Mouseover tooltip now displays "The user name of your RADIUS user."

- Fixed an issue with password requirement validation. When a password policy specified that the username must not be included in the password, the validation was too aggressive. Now at least three consecutive characters in the password must match a section of the username for a password to be rejected.
- Fixed an issue that caused locks to hang in tbDatabaseCache.
- Fixed an issue where an account is not being used as a "Privileged Secrets for Scripts" when importing for discovery for RPC and heartbeat.
- Fixed an issue where RDP proxy may experience intermittent connectivity when using the Mac version of Connection Manager.
- Fixed an issue causing Secret Server-BWSR errors from UDP Syslog datagrams exceeding RFC3164/5424 specifications. Note that TCP or SecureTCP, is preferred over UDP for reliability purposes.
- Fixed an issue when editing the default password requirement when using a language other than English. It set the default to false.
- Fixed an issue when syncing AzureAD where deleting a sync group in AzureAD would result in members of all groups being synchronized.

Heartbeat, Distributed Engines, and RPC

- Fixed an issue that resulted in a "Phantom.JS.exe not found" error when trying to run a heartbeat on a Web password with a distributed engine.
- Fixed an issue where password changing for secrets was not being triggered after a forced check-in.
- Fixed an issue when creating custom dependencies that produced an invalid error when adding a parent field of 'Machine.'
- Fixed an issue when setting the auto change schedule on a secret policy to monthly that caused a UI issue that prevented viewing the Remote Password Changing tab.
- Fixed an issue where a "Last Heartbeat Status" error was displayed in the UI that indicated the user's Web browser stopped polling the server. An improved error message is provided in cases where polling timeout still occurs.
- Fixed an issue with password changing for SAP secrets that caused failures at password change due to an issue with the check-in.

  For SAP servers where rfc/reject_expired_passwd = 1, a new option was added to the Advanced Settings in the SAP password changer. This new option, "Use Single Destination (SAP)" is false by default, but, when set to true, it allows privileged password changing to succeed on these servers by using the privileged credentials in both steps of the privileged password change.

  For servers where rfc/reject_expired_passwd = 0, this option may be set to true or false and password changing will succeed.
- Fixed an issue where previously inactive custom scripts were still displayed as available for RPC.
- Fixed an issue where a secret policy which enforces 'View Password Requires Edit' prevents API access to the view the password.
- Fixed an issue that displayed the distributed engine count incorrectly based on previously deleted distributed engines.

- Fixed an issue that could prevent the privileged account password from changing by a distributed engine if a child secret were deleted while it was being processed for RPC.
- Fixed an issue preventing a scheduled password change if the secret template expiration is disabled.
- Fixed an issue producing an error during a distributed engine install when the hostname was more than 50 characters.
- Fixed an issue for the ODBC password changer with verification opening the connection but not executing custom commands. The fix allows for: CheckFor, CheckContains, and CheckNotContains to be used in custom commands. Additional logging has also been added when verbose logging is enabled.
- Fixed an issue with RPC on scheduled tasks that have a secret ID on a different trusted domain.
- Fixed an issue where the distributed engine automatic upgrade would fail if the OS account HKU{GUID} entry does not have permission to stop the distributed engine service.

Installation, Upgrade, and Uninstall

- Fixed an issue where ThycoticSetup.exe failed to install SQL Express when selected for the installation.
- Fixed an issue producing an error during a distributed engine install when the hostname was more than 50 characters.

Launchers

Fixed an issue with SAP launcher where record "Additional Processes" stops recording if it was opened from another process within "Process Arguments" and that process is closed.

Reports

Fixed an issue where duplicate report emails were received when a report schedule was not removed after a report with the same name was deleted. The fix does not allow two active reports to have the same report name when creating or updating a report.

Secret Server Cloud

- Fixed an issue where Secret Server Cloud users were presented with a misleading tooltip for user synchronization interval options in directory services configuration. An improved tooltip has been provided to inform the user the limitations of the configuration more accurately in Secret Server Cloud.
- Fixed an issue in Secret Server Cloud where personal folders were not being created for new users created through Delinea One.
- Fixed an issue for Secret Server Cloud where after entering the PIN for Duo and attempting login, Secret Server would activate the Duo Push button in the UI, rather than proceeding with the login.

Secrets, Policies, and Templates

- Fixed an issue by adding more robust permission checks to determine when template conversion options should be available to users, rather than displaying generic access denied errors when users attempted making conversions on a template without the right user roles.

- Fixed an issue by providing additional detail in a warning message to alert the users when duplicating templates that changes must be applied to all secrets intended to use the new template.
- Fixed an issue when setting the auto change schedule on a secret policy to monthly that caused a UI issue that prevented viewing the Remote Password Changing tab.
- Fixed an issue that prevented record retrieval on secret permissions if the application account has owner permissions at the secret level. Now, if you have view access to the secret in the filter you can retrieve the permissions. Queries that do not have secret ID still require the secret owner.
- Fixed an issue where a secret template with a duplicate name in fields would cause a failure when exported, due to the field name match. We recommend using an available slug on the XML for handling fields which have the same name.
- Fixed an issue in the UI when searching of the secrets grid where the search text box covered the first result in the grid.
- Fixed and improved reference labeling to secrets when a secret template changes on deleted secrets. A (deleted) reference is added to secrets used in a secret policy that have been deleted. Added a note to let the user know that deleted secrets are included in the secret count.
- Fixed an issue preventing a scheduled password change if the secret template expiration is disabled.
- Fixed an issue preventing applying a secret policy without access to the required privileged account. Users can now assign a new role that allows users with "list" permissions on the secret to pass the secret access permissions check for privileged and associated secrets when the privileged or associated secrets are enforced by secret policy.

Session Recording

- Fixed an issue causing Session Recording failures to record multiple sessions to the same target with MobaXTerm.
- Fixed an issue where older session recording keystroke logs were not deleted after the configured retention period. Session recording using SSH or RDP Proxy to capture keystroke metadata was not running the cleanup job after the data retention period expired.
- Fixed an issue where searching for recorded sessions could produce an execution timeout error.
- Fixed an issue with the SAP launcher only recording part of the screen.

SSH Proxy and Terminal

- Fixed an issue where SSH proxy sessions via SSH terminal would close within approximately five minutes after launch.
- Fixed an issue where when a distributed engine was used as a proxy server the SSH terminal gave an invalid error during authentication for public keys.
- Fixed an issue when launching SSH secrets via SSH terminal that have 'check out' and 'change on check in' enabled. When exiting the launched session, the secret does not check in.
- Fixed an issue with proxied SSH custom launchers using a "Connect As" secret failing to launch successfully.

- Fixed an issue using private ECDSA SSH keys generated in the OpenSSH format that produced an error when uploading the key on a SSH key template.
- Fixed an issue that caused premature closure of proxied SSH processes when opening multiple sessions of MobaXterm.
- Fixed an issue where the SSH proxy tunnel would not take $password from Xshell client on a custom launcher.
- Fixed an issue where the supported SSH key exchange algorithms would fail to negotiate (ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521).
- Fixed an issue where the keyboard did not operate for users on SSH RDP PuTTY sessions via proxy.

Users and Groups

Fixed an issue with the "Users Migrate to AD" function producing an invalid error when canceling.

Web Password Filler

Fixed an issue where Web Password Filler did not prompt token generation for users on Windows 10 version 20H2.

Fixes made since Early Adopter Version 11.0.000000

- Fixed a high, CVSS 8.6, security issue. See the important note at the top of these release notes.
- Fixed a critical, CVSS 9.9, security issue. It impacts Secret Server 10.9.000032 to 11.0.000006. See the important note at the top of the 11.000007 release notes.
- Fixed an issue with secret access requests not allowing the administrator to approve via the inbox.
- Fixed an issue where SSH blocklist caused user keystrokes to be echoed back in the session.
- Fixed an issue with SSH blocklist that caused all su sessions to quit when issuing an exit command.
- Fixed an issue with SSH blocklist where the control-c input does not quit running the program when SSH blocklist is enabled.
- Fixed an issue with SSH blocklist where ESC input on a session with SSH blocklist terminates the session.
- Fixed an issue with SSH blocklist where all su sessions terminate when issuing an exit command.
- Fixed an issue with the lists feature where a user with own or edit permission on a secret were not given access to add or remove lists.
- Fixed date parsing logic for license dates. License dates are in U.S. date format, but default parsing logic uses local-server time-format settings to parse the date and could fail if the expected format is in a different order, for example, in the U.K. where the format is dd/mm/yyyy instead of mm/dd/yyyy. We forced the parser to use the U.S. date format.

Pending Deprecations

This section describes planned feature or platform-support deprecations in Secret Server.

- Internet Explorer 11. Support for Internet Explorer 11 ends on 31 August 2021. Secret Server releases after that date will not support Internet Explorer 11.

- Secret Server Classic UI. The Classic UI option in Secret Server is scheduled to be removed in Q1 2022. After that time, the New UI will be the only available UI option in Secret Server.

Secret Server: 11.0.000007 Release Notes

Release date:

- September 7, 2021 (on-premises only)
- September 11, 2021 (Cloud: US)

> This release addresses a security issue that was recently discovered during third-party penetration testing. See "Secret Server 11.1.000007 Security Update" below for details.

> If you installed Secret Server as your default or top-level website and you have Privilege Manager (PM) and Secret Server installed together, you may experience the following issues after upgrading to .NET Framework 4.8:
>
> - PM agents will not register.
> - When a PM agent updates itself (using the agent utility), it states that there are zero policies to download.
>
> If you believe this scenario applies to you, please contact Delinea Support **before** performing a .NET, Secret Server, or PM upgrade.

Secret Server 11.1.000007 Security Update

This release addresses a security issue that was recently discovered during third-party penetration testing. The security issue is rated High severity on the standard CVSS scale and impacts all previous versions of Secret Server On-Premise. It impacts Secret Server**10.9.000032** to **11.0.000006**. We recommend all affected Secret Server on-premise customers upgrade immediately to version **11.0.000007**.

It does not impact Secret Server Cloud and there were no version updates to Secret Server Cloud.

CVSS Summary:

NTLM Hash Leak:

Short Description: A highly privileged user on the same network as Secret Server could potentially cause an unauthorized NTLM hash leak. The vulnerability is given a score of 7.4 by CVSS v3.1 Vector AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:H.

> Delinea thanks Filip Magic and Florian Coman from TDC NET Security for identifying the security issue leading to the 11.0.000007 release.

> We deployed a security patch for this same issue to all Secret Server Cloud regions on September 3rd. The update was independent of the 11.000007 feature update, which occurs on September 11.

Known Issues

Connection Manager Version 1.6.2 Required for Customers Using Secret Server 11.0

Details:

- Intermittent Issue: On a macOS, when run in protocol handler mode, copy and paste does not work. The app does not crash but the copy and paste action is not successful. If the app is “unlocked,” copy, cut, and paste work. For your Secret Server connection to “unlock” it, expand the left navigation panel, and double click on the “lock” icon.
- Intermittent Issue On a macOS, while connected to any Secret Server, if the user edits the same Secret Server name (via the Secret Server connections menu), the app could get stuck in a “connecting…” state.
- Intermittent Issue: On a macOS, some users may experience issues connecting to Secret Server via a Web login (SAML) where the connection stays in a “connecting…” state.
- Upgrade Issues: In some situations, the automatic upgrade may not run the installer after downloading the MSI or package files. On macOS, when upgrading from v1.6.0, the automatic update fails to launch the installer. On Windows, the auto update may not launch the installer. In that case, please download and run the installer manually or via a software deployment tool.

  Installer download links:

  - macOS
  - Windows

New Features and Enhancements

API

Secrets are now searchable by the full path. This removes the need to know the ID to locate the intended secret.

Encrypted Secret Export

Secret Server now allows you to schedule encrypted exports of secret data to external storage.

General

- Updated ESXi integration support via an update of PowerCLI 12.
- Updated PuTTY to version .074.
- Added SNC support for communication between SAP and Secret Server for heartbeat and remote password changing.
- Added support for metadata filters on event pipelines.
- Added the ability to intercept a sudo or su command in a proxied session and inject the session password directly from Secret Server, not disclosing the password to the user.
- Added support for displaying both username and display name in audit logs.
- Added SMB fallback support for local Windows account heartbeats.

- Added a discovery threshold configuration that controls when a dependency is deactivated if it is not found in a scan. The threshold is set to the times a dependency can be missing prior to deactivation. It can also be set to "never" to prevent deactivation.

Inbox

The inbox now provides a customizable toolset to manage how email and notifications are sent and received by users. Inbox allows for configuration of notification scheduling, collecting notifications into digests, creation of message templates and rules, and more.

Lists

Secrets now support configurable data lists. Users with the "apply lists to secrets via secret policies" permission can create a list. This provides an easy mechanism for secret owners to simply choose from provided lists, such as a list of machines the user can select. Additionally, lists can be used for allow and blocklists allowing for control over what the secret owner can access.

Proxy Generic Connections

Secret Server can now securely tunnel a connection to servers operating on a variety of protocols.

Secret Erase

Users with the permissions to use the secret erase feature can permanently erase data from a secret. This provides Secret Server a method to purge secret data without reconciliation of the erased data. The existing "delete" function (now called "deactivate") allows you to "undelete" (reactivate). Secret erase can be audited as an event.

SSH Key Discovery

Secret Server discovery can now discover SSH public keys by scanning key locations on Linux and Unix servers.

User Interface

- Added dependency status reports. The reports are:
  - Overview: Shows how many dependencies failed, succeeded, and were not run.
  - Status: Shows how many dependencies failed, succeeded, and were not run by clickable secret, secret dependency group, and site.
  - Failed by secret: Shows doughnut graphic with secret and fail count.
  - Not run by secret: Shows the not run count by secret.
- Renamed the "Viewing Password Requires Edit" setting to "Hide Launcher Password" in the secret policies editor to improve clarity.

Bug Fixes

Note: The same line item may appear in more than one section when it applies to both.

Access Requests and Secret Workflows

- Fixed an issue where the "Revoke" button was being displayed in the UI after a request had already expired. The button should not be available to users past expiration.
- Fixed an issue where a "No Permission" page is displayed incorrectly when checking-in a secret with "change on check-in" enabled.

Alerts, Events, and Logging

- Fixed an issue where if the browser is set to a non-English language, a failed login attempt causes invalid system logging. The fix changes the character type from text to nvchar to produce successful logging entries. For this data type conversion, we recommend that customers with lots of data may want to run the SQL delta ahead of an upgrade to prevent potential SQL timeouts during upgrade.

API

- Fixed an issue where API was returning null for createDate on SecretSummary queries.
- Fixed an issue that resulted in receiving a 500 error when using the REST API to enable check-in on a secret using the PATCH method /secrets/{id}/security-checkout.
- Fixed an issue where the REST API was returning "Object reference not set to an instance of an object." against the /directory-services/domains/{domainid}/group.
- Fixed an issue for the API endpoint, workflow, or template throwing an exception when the take parameter was not specified.
- Fixed a POST issue in secret-templates REST endpoint that did not properly validate the editablePermission property on fields.
- Fixed an issue where a secret policy which enforces 'View Password Requires Edit' prevents API access to the view the password.

Authentication, Login, and Directory Services

- Fixed an issue with post authentication for SAML.
- Fixed an issue where if the browser is set to a non-English language, a failed login attempt causes invalid system logging. The fix changes the character type from text to nvchar to produce successful logging entries. For this data type conversion, we recommend that customers with lots of data may want to run the SQL delta ahead of an upgrade to prevent potential SQL timeouts during upgrade.
- Fixed an issue where when a distributed engine was used as a proxy server the SSH terminal gave an invalid error during authentication for public keys.
- Fixed an issue where occasionally users that synced via Active Directory were not getting assigned default role (s).
- Fixed an issue with Active Directory synchronization throwing a credential validation error when the sync credentials are from another domain after creating relationship between domains.
- Added a notice in the UI for directory services group sync search results being limited to 1000 groups.

- Fixed an issue where Active Directory group names were showing with their pre-Windows 2000 name instead of the group name.
- Fixed an issue where Secret Server Cloud users were presented with a misleading tooltip for user synchronization interval options in directory services configuration. An improved tooltip has been provided to inform the user the limitations of the configuration more accurately in Secret Server Cloud.

Discovery

- Fixed an issue in discovery when using LDAPS where discovery attempts to use port 389 despite the user having selected LDAPS port 636.
- Fixed an issue with saving IP addresses additions to the manual host range within the discovery scanner. A message is now provided to inform the user that the manual host ranges should be set on the discovery source scanner, rather than as a default setting on the scanner itself.
- Fixed an issue that could occasionally cause dependencies to get disabled by a computer discovery scan.
- Fixed a discovery issue that could cause background processes to stop.
- Fixed an issue where discovery was retaining too many historical records for each host. We added an advanced configuration setting to define the number of records.
- Fixed a discovery issue where when the discoverysourceid was equal to 9 or 10, after the domain scan completed to get machines, the Service Account tab in the UI would not load re-scan buttons.
- Fixed an issue where the schedule task scanner did not work for the domain discovery source with two UPNs.
- Fixed an issue with discovery for scheduled task scanning across multiple trusted domains.
- Fixed a discovery issue where similar OU names resulted in discovery importing accounts from an OU with a similar name to the targeted OU.

Event Subscriptions and Pipelines

- Fixed an issue that could be encountered with event subscriptions after converting a secret's template. An improved message is now provided in the UI when converting secret templates to inform the user what actions needs to take place to prevent the issue.
- Fixed an issue with event pipelines receiving an application error when trying to add tasks in a policy if the policy has a deactivated pipeline.
- Fixed an event subscriptions issue of not initiating for engine events.

Folders

- Fixed an issue when editing folder permissions and receiving no search results for groups that contained a '+' in the group name.
- Fixed an issue that prevented the ability to sort folders by alphabetical order on name in the Favorites tab.
- Fixed an issue where the folder name may not display in the secret grid.
- Fixed an issue where right clicking secret folders in the UI did not have the "View Details" option available for users with view permissions.

- Fixed an issue where editing folder sharing permissions removed restricted template options in the new UI.
- Fixed an issue that caused an application error when changing a folder's secret permissions to "none."

General

- Fixed an issue that caused a 500 error if certain password sequences were used when password validation settings were enabled to prevent password sequences.
- Fixed an issue that prevented content deletion in metadata fields.
- Fixed an issue where pressing enter on the keyboard when using the main application search box did not display results in the secret grid but only in the dropdown.
- Fixed an issue where the "Copy to Clipboard" button was not appearing on fields that contained a character count beyond the max field character display.
- Fixed an issue where the "Options" menu on pages is automatically hidden in the UI if no options are present to display to the user.
- Fixed an issue where sessions created through the Secret Server terminal "Launch" command were not getting returned by session search when filtering by secret name, secret field value, or user.
- Fixed an issue where on the home page engine status widget was not displaying the correct status in the "Connection Status 1" column. In the fix, this column has been renamed to 'Activation Status'.
- Fixed an invalid audit message entry when a user initiates "Change Password Now" using the "Randomly Generated" option.
- Fixed an issue where changing the name of a launcher did not update the name of the launcher on the secret.
- Fixed a mouseover label on "RADIUS User Name" on the Users page. Mouseover tooltip now displays "The user name of your RADIUS user."
- Fixed an issue with password requirement validation. When a password policy specified that the username must not be included in the password, the validation was too aggressive. Now at least three consecutive characters in the password must match a section of the username for a password to be rejected.
- Fixed an issue that caused locks to hang in tbDatabaseCache.
- Fixed an issue where an account is not being used as a "Privileged Secrets for Scripts" when importing for discovery for RPC and heartbeat.
- Fixed an issue where RDP proxy may experience intermittent connectivity when using the Mac version of Connection Manager.
- Fixed an issue causing Secret Server-BWSR errors from UDP Syslog datagrams exceeding RFC3164/5424 specifications. Note that TCP or SecureTCP, is preferred over UDP for reliability purposes.
- Fixed an issue when editing the default password requirement when using a language other than English. It set the default to false.
- Fixed an issue when syncing AzureAD where deleting a sync group in AzureAD would result in members of all groups being synchronized.

Heartbeat, Distributed Engines, and RPC

- Fixed an issue that resulted in a "Phantom.JS.exe not found" error when trying to run a heartbeat on a Web password with a distributed engine.
- Fixed an issue where password changing for secrets was not being triggered after a forced check-in.
- Fixed an issue when creating custom dependencies that produced an invalid error when adding a parent field of 'Machine.'
- Fixed an issue when setting the auto change schedule on a secret policy to monthly that caused a UI issue that prevented viewing the Remote Password Changing tab.
- Fixed an issue where a "Last Heartbeat Status" error was displayed in the UI that indicated the user's Web browser stopped polling the server. An improved error message is provided in cases where polling timeout still occurs.
- Fixed an issue with password changing for SAP secrets that caused failures at password change due to an issue with the check-in.

  For SAP servers where rfc/reject_expired_passwd = 1, a new option was added to the Advanced Settings in the SAP password changer. This new option, "Use Single Destination (SAP)" is false by default, but, when set to true, it allows privileged password changing to succeed on these servers by using the privileged credentials in both steps of the privileged password change.

  For servers where rfc/reject_expired_passwd = 0, this option may be set to true or false and password changing will succeed.
- Fixed an issue where previously inactive custom scripts were still displayed as available for RPC.
- Fixed an issue where a secret policy which enforces 'View Password Requires Edit' prevents API access to the view the password.
- Fixed an issue that displayed the distributed engine count incorrectly based on previously deleted distributed engines.
- Fixed an issue that could prevent the privileged account password from changing by a distributed engine if a child secret were deleted while it was being processed for RPC.
- Fixed an issue preventing a scheduled password change if the secret template expiration is disabled.
- Fixed an issue producing an error during a distributed engine install when the hostname was more than 50 characters.
- Fixed an issue for the ODBC password changer with verification opening the connection but not executing custom commands. The fix allows for: CheckFor, CheckContains, and CheckNotContains to be used in custom commands. Additional logging has also been added when verbose logging is enabled.
- Fixed an issue with RPC on scheduled tasks that have a secret ID on a different trusted domain.
- Fixed an issue where the distributed engine automatic upgrade would fail if the OS account HKU{GUID} entry does not have permission to stop the distributed engine service.

Installation, Upgrade, and Uninstall

- Fixed an issue where ThycoticSetup.exe failed to install SQL Express when selected for the installation.
- Fixed an issue producing an error during a distributed engine install when the hostname was more than 50 characters.

Launchers

Fixed an issue with SAP launcher where record "Additional Processes" stops recording if it was opened from another process within "Process Arguments" and that process is closed.

Reports

Fixed an issue where duplicate report emails were received when a report schedule was not removed after a report with the same name was deleted. The fix does not allow two active reports to have the same report name when creating or updating a report.

Secret Server Cloud

- Fixed an issue where Secret Server Cloud users were presented with a misleading tooltip for user synchronization interval options in directory services configuration. An improved tooltip has been provided to inform the user the limitations of the configuration more accurately in Secret Server Cloud.
- Fixed an issue in Secret Server Cloud where personal folders were not being created for new users created through Delinea One.
- Fixed an issue for Secret Server Cloud where after entering the PIN for Duo and attempting login, Secret Server would activate the Duo Push button in the UI, rather than proceeding with the login.

Secrets, Policies, and Templates

- Fixed an issue by adding more robust permission checks to determine when template conversion options should be available to users, rather than displaying generic access denied errors when users attempted making conversions on a template without the right user roles.
- Fixed an issue by providing additional detail in a warning message to alert the users when duplicating templates that changes must be applied to all secrets intended to use the new template.
- Fixed an issue when setting the auto change schedule on a secret policy to monthly that caused a UI issue that prevented viewing the Remote Password Changing tab.
- Fixed an issue that prevented record retrieval on secret permissions if the application account has owner permissions at the secret level. Now, if you have view access to the secret in the filter you can retrieve the permissions. Queries that do not have secret ID still require the secret owner.
- Fixed an issue where a secret template with a duplicate name in fields would cause a failure when exported, due to the field name match. We recommend using an available slug on the XML for handling fields which have the same name.
- Fixed an issue in the UI when searching of the secrets grid where the search text box covered the first result in the grid.

- Fixed and improved reference labeling to secrets when a secret template changes on deleted secrets. A (deleted) reference is added to secrets used in a secret policy that have been deleted. Added a note to let the user know that deleted secrets are included in the secret count.
- Fixed an issue preventing a scheduled password change if the secret template expiration is disabled.
- Fixed an issue preventing applying a secret policy without access to the required privileged account. Users can now assign a new role that allows users with "list" permissions on the secret to pass the secret access permissions check for privileged and associated secrets when the privileged or associated secrets are enforced by secret policy.

Session Recording

- Fixed an issue causing Session Recording failures to record multiple sessions to the same target with MobaXTerm.
- Fixed an issue where older session recording keystroke logs were not deleted after the configured retention period. Session recording using SSH or RDP Proxy to capture keystroke metadata was not running the cleanup job after the data retention period expired.
- Fixed an issue where searching for recorded sessions could produce an execution timeout error.
- Fixed an issue with the SAP launcher only recording part of the screen.

SSH Proxy and Terminal

- Fixed an issue where SSH proxy sessions via SSH terminal would close within approximately five minutes after launch.
- Fixed an issue where when a distributed engine was used as a proxy server the SSH terminal gave an invalid error during authentication for public keys.
- Fixed an issue when launching SSH secrets via SSH terminal that have 'check out' and 'change on check in' enabled. When exiting the launched session, the secret does not check in.
- Fixed an issue with proxied SSH custom launchers using a "Connect As" secret failing to launch successfully.
- Fixed an issue using private ECDSA SSH keys generated in the OpenSSH format that produced an error when uploading the key on a SSH key template.
- Fixed an issue that caused premature closure of proxied SSH processes when opening multiple sessions of MobaXterm.
- Fixed an issue where the SSH proxy tunnel would not take $password from Xshell client on a custom launcher.
- Fixed an issue where the supported SSH key exchange algorithms would fail to negotiate (ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521).
- Fixed an issue where the keyboard did not operate for users on SSH RDP PuTTY sessions via proxy.

Users and Groups

Fixed an issue with the "Users Migrate to AD" function producing an invalid error when canceling.

Web Password Filler

Fixed an issue where Web Password Filler did not prompt token generation for users on Windows 10 version 20H2.

Fixes made since Early Adopter Version 11.0.000000

- Fixed a critical, CVSS 9.9, security issue. It impacts Secret Server 10.9.000032 to 11.0.000006. See the important note at the top of these release notes.
- Fixed an issue with secret access requests not allowing the administrator to approve via the inbox.
- Fixed an issue where SSH blocklist caused user keystrokes to be echoed back in the session.
- Fixed an issue with SSH blocklist that caused all su sessions to quit when issuing an exit command.
- Fixed an issue with SSH blocklist where the control-c input does not quit running the program when SSH blocklist is enabled.
- Fixed an issue with SSH blocklist where ESC input on a session with SSH blocklist terminates the session.
- Fixed an issue with SSH blocklist where all su sessions terminate when issuing an exit command.
- Fixed an issue with the lists feature where a user with own or edit permission on a secret were not given access to add or remove lists.
- Fixed date parsing logic for license dates. License dates are in U.S. date format, but default parsing logic uses local-server time-format settings to parse the date and could fail if the expected format is in a different order, for example, in the U.K. where the format is dd/mm/yyyy instead of mm/dd/yyyy. We forced the parser to use the U.S. date format.

Pending Deprecations

This section describes planned feature or platform-support deprecations in Secret Server.

- Internet Explorer 11. Support for Internet Explorer 11 ends on 31 August 2021. Secret Server releases after that date will not support Internet Explorer 11.
- Secret Server Classic UI. The Classic UI option in Secret Server is scheduled to be removed in Q1 2022. After that time, the New UI will be the only available UI option in Secret Server.

Secret Server: 11.0.000006 Release Notes

Release date:

- July 21, 2021 (early release, on-premises only)
- August 17, 2021 (on-premises version)
- August 28, 2021 (Cloud: CA, SG, AU, EU)
- September 11, 2021(Cloud: US)

Note: These dates are tentative. Please see the main release note page for the actually dates as they happen.

Important: If you are in a Secret Server Cloud region that has not yet received the 11.0.x release, please use the "Secret Server: 10.9.000064 Release Notes" on page 1788 notes instead.

Important: If you installed Secret Server as your default or top-level website and you have Privilege Manager (PM) and Secret Server installed together, you may experience the following issues after upgrading to .NET Framework 4.8:

- PM agents will not register.
- When a PM agent updates itself (using the agent utility), it states that there are zero policies to download.

If you believe this scenario applies to you, please contact Delinea Support **before** performing a .NET, Secret Server, or PM upgrade.

Known Issues

Connection Manager Version 1.6.2 Required for Customers Using Secret Server 11.0

Details:

- Intermittent Issue: On a macOS, when run in protocol handler mode, copy and paste does not work. The app does not crash but the copy and paste action is not successful. If the app is "unlocked," copy, cut, and paste work. For your Secret Server connection to "unlock" it, expand the left navigation panel, and double click on the "lock" icon.
- Intermittent Issue On a macOS, while connected to any Secret Server, if the user edits the same Secret Server name (via the Secret Server connections menu), the app could get stuck in a "connecting..." state.
- Intermittent Issue: On a macOS, some users may experience issues connecting to Secret Server via a Web login (SAML) where the connection stays in a "connecting..." state.
- Upgrade Issues: In some situations, the automatic upgrade may not run the installer after downloading the MSI or package files. On macOS, when upgrading from v1.6.0, the automatic update fails to launch the installer. On Windows, the auto update may not launch the installer. In that case, please download and run the installer manually or via a software deployment tool.

  Installer download links:

  - macOS
  - Windows

New Features and Enhancements

API

Secrets are now searchable by the full path. This removes the need to know the ID to locate the intended secret.

Encrypted Secret Export

Secret Server now allows you to schedule encrypted exports of secret data to external storage.

General

- Updated ESXi integration support via an update of PowerCLI 12.
- Updated PuTTY to version .074.
- Added SNC support for communication between SAP and Secret Server for heartbeat and remote password changing.
- Added support for metadata filters on event pipelines.

- Added the ability to intercept a sudo or su command in a proxied session and inject the session password directly from Secret Server, not disclosing the password to the user.
- Added support for displaying both username and display name in audit logs.
- Added SMB fallback support for local Windows account heartbeats.
- Added a discovery threshold configuration that controls when a dependency is deactivated if it is not found in a scan. The threshold is set to the times a dependency can be missing prior to deactivation. It can also be set to "never" to prevent deactivation.

Inbox

The inbox now provides a customizable toolset to manage how email and notifications are sent and received by users. Inbox allows for configuration of notification scheduling, collecting notifications into digests, creation of message templates and rules, and more.

Lists

Secrets now support configurable data lists. Users with the "apply lists to secrets via secret policies" permission can create a list. This provides an easy mechanism for secret owners to simply choose from provided lists, such as a list of machines the user can select. Additionally, lists can be used for allow and blocklists allowing for control over what the secret owner can access.

Proxy Generic Connections

Secret Server can now securely tunnel a connection to servers operating on a variety of protocols.

Secret Erase

Users with the permissions to use the secret erase feature can permanently erase data from a secret. This provides Secret Server a method to purge secret data without reconciliation of the erased data. The existing "delete" function (now called "deactivate") allows you to "undelete" (reactivate). Secret erase can be audited as an event.

SSH Key Discovery

Secret Server discovery can now discover SSH public keys by scanning key locations on Linux and Unix servers.

User Interface

- Added dependency status reports. The reports are:
  - Overview: Shows how many dependencies failed, succeeded, and were not run.
  - Status: Shows how many dependencies failed, succeeded, and were not run by clickable secret, secret dependency group, and site.
  - Failed by secret: Shows doughnut graphic with secret and fail count.
  - Not run by secret: Shows the not run count by secret.
- Renamed the "Viewing Password Requires Edit" setting to "Hide Launcher Password" in the secret policies editor to improve clarity.

Bug Fixes

Note: The same line item may appear in more than one section when it applies to both.

Access Requests and Secret Workflows

- Fixed an issue where the "Revoke" button was being displayed in the UI after a request had already expired. The button should not be available to users past expiration.
- Fixed an issue where a "No Permission" page is displayed incorrectly when checking-in a secret with "change on check-in" enabled.

Alerts, Events, and Logging

- Fixed an issue where if the browser is set to a non-English language, a failed login attempt causes invalid system logging. The fix changes the character type from text to nvchar to produce successful logging entries. For this data type conversion, we recommend that customers with lots of data may want to run the SQL delta ahead of an upgrade to prevent potential SQL timeouts during upgrade.

API

- Fixed an issue where API was returning null for createDate on SecretSummary queries.
- Fixed an issue that resulted in receiving a 500 error when using the REST API to enable check-in on a secret using the PATCH method /secrets/{id}/security-checkout.
- Fixed an issue where the REST API was returning "Object reference not set to an instance of an object." against the /directory-services/domains/{domainid}/group.
- Fixed an issue for the API endpoint, workflow, or template throwing an exception when the take parameter was not specified.
- Fixed a POST issue in secret-templates REST endpoint that did not properly validate the editablePermission property on fields.
- Fixed an issue where a secret policy which enforces 'View Password Requires Edit' prevents API access to the view the password.

Authentication, Login, and Directory Services

- Fixed an issue with post authentication for SAML.
- Fixed an issue where if the browser is set to a non-English language, a failed login attempt causes invalid system logging. The fix changes the character type from text to nvchar to produce successful logging entries. For this data type conversion, we recommend that customers with lots of data may want to run the SQL delta ahead of an upgrade to prevent potential SQL timeouts during upgrade.
- Fixed an issue where when a distributed engine was used as a proxy server the SSH terminal gave an invalid error during authentication for public keys.
- Fixed an issue where occasionally users that synced via Active Directory were not getting assigned default role (s).

- Fixed an issue with Active Directory synchronization throwing a credential validation error when the sync credentials are from another domain after creating relationship between domains.
- Added a notice in the UI for directory services group sync search results being limited to 1000 groups.
- Fixed an issue where Active Directory group names were showing with their pre-Windows 2000 name instead of the group name.
- Fixed an issue where Secret Server Cloud users were presented with a misleading tooltip for user synchronization interval options in directory services configuration. An improved tooltip has been provided to inform the user the limitations of the configuration more accurately in Secret Server Cloud.

Discovery

- Fixed an issue in discovery when using LDAPS where discovery attempts to use port 389 despite the user having selected LDAPS port 636.
- Fixed an issue with saving IP addresses additions to the manual host range within the discovery scanner. A message is now provided to inform the user that the manual host ranges should be set on the discovery source scanner, rather than as a default setting on the scanner itself.
- Fixed an issue that could occasionally cause dependencies to get disabled by a computer discovery scan.
- Fixed a discovery issue that could cause background processes to stop.
- Fixed an issue where discovery was retaining too many historical records for each host. We added an advanced configuration setting to define the number of records.
- Fixed a discovery issue where when the discoverysourceid was equal to 9 or 10, after the domain scan completed to get machines, the Service Account tab in the UI would not load re-scan buttons.
- Fixed an issue where the schedule task scanner did not work for the domain discovery source with two UPNs.
- Fixed an issue with discovery for scheduled task scanning across multiple trusted domains.
- Fixed a discovery issue where similar OU names resulted in discovery importing accounts from an OU with a similar name to the targeted OU.

Event Subscriptions and Pipelines

- Fixed an issue that could be encountered with event subscriptions after converting a secret's template. An improved message is now provided in the UI when converting secret templates to inform the user what actions needs to take place to prevent the issue.
- Fixed an issue with event pipelines receiving an application error when trying to add tasks in a policy if the policy has a deactivated pipeline.
- Fixed an event subscriptions issue of not initiating for engine events.

Folders

- Fixed an issue when editing folder permissions and receiving no search results for groups that contained a '+' in the group name.
- Fixed an issue that prevented the ability to sort folders by alphabetical order on name in the Favorites tab.
- Fixed an issue where the folder name may not display in the secret grid.

- Fixed an issue where right clicking secret folders in the UI did not have the "View Details" option available for users with view permissions.
- Fixed an issue where editing folder sharing permissions removed restricted template options in the new UI.
- Fixed an issue that caused an application error when changing a folder's secret permissions to "none."

General

- Fixed an issue that caused a 500 error if certain password sequences were used when password validation settings were enabled to prevent password sequences.
- Fixed an issue that prevented content deletion in metadata fields.
- Fixed an issue where pressing enter on the keyboard when using the main application search box did not display results in the secret grid but only in the dropdown.
- Fixed an issue where the "Copy to Clipboard" button was not appearing on fields that contained a character count beyond the max field character display.
- Fixed an issue where the "Options" menu on pages is automatically hidden in the UI if no options are present to display to the user.
- Fixed an issue where sessions created through the Secret Server terminal "Launch" command were not getting returned by session search when filtering by secret name, secret field value, or user.
- Fixed an issue where on the home page engine status widget was not displaying the correct status in the "Connection Status 1" column. In the fix, this column has been renamed to 'Activation Status'.
- Fixed an invalid audit message entry when a user initiates "Change Password Now" using the "Randomly Generated" option.
- Fixed an issue where changing the name of a launcher did not update the name of the launcher on the secret.
- Fixed a mouseover label on "RADIUS User Name" on the Users page. Mouseover tooltip now displays "The user name of your RADIUS user."
- Fixed an issue with password requirement validation. When a password policy specified that the username must not be included in the password, the validation was too aggressive. Now at least three consecutive characters in the password must match a section of the username for a password to be rejected.
- Fixed an issue that caused locks to hang in tbDatabaseCache.
- Fixed an issue where an account is not being used as a "Privileged Secrets for Scripts" when importing for discovery for RPC and heartbeat.
- Fixed an issue where RDP proxy may experience intermittent connectivity when using the Mac version of Connection Manager.
- Fixed an issue causing Secret Server-BWSR errors from UDP Syslog datagrams exceeding RFC3164/5424 specifications. Note that TCP or SecureTCP, is preferred over UDP for reliability purposes.
- Fixed an issue when editing the default password requirement when using a language other than English. It set the default to false.
- Fixed an issue when syncing AzureAD where deleting a sync group in AzureAD would result in members of all groups being synchronized.

Heartbeat, Distributed Engines, and RPC

- Fixed an issue that resulted in a "Phantom.JS.exe not found" error when trying to run a heartbeat on a Web password with a distributed engine.
- Fixed an issue where password changing for secrets was not being triggered after a forced check-in.
- Fixed an issue when creating custom dependencies that produced an invalid error when adding a parent field of 'Machine.'
- Fixed an issue when setting the auto change schedule on a secret policy to monthly that caused a UI issue that prevented viewing the Remote Password Changing tab.
- Fixed an issue where a "Last Heartbeat Status" error was displayed in the UI that indicated the user's Web browser stopped polling the server. An improved error message is provided in cases where polling timeout still occurs.
- Fixed an issue with password changing for SAP secrets that caused failures at password change due to an issue with the check-in.

  For SAP servers where rfc/reject_expired_passwd = 1, a new option was added to the Advanced Settings in the SAP password changer. This new option, "Use Single Destination (SAP)" is false by default, but, when set to true, it allows privileged password changing to succeed on these servers by using the privileged credentials in both steps of the privileged password change.

  For servers where rfc/reject_expired_passwd = 0, this option may be set to true or false and password changing will succeed.
- Fixed an issue where previously inactive custom scripts were still displayed as available for RPC.
- Fixed an issue where a secret policy which enforces 'View Password Requires Edit' prevents API access to the view the password.
- Fixed an issue that displayed the distributed engine count incorrectly based on previously deleted distributed engines.
- Fixed an issue that could prevent the privileged account password from changing by a distributed engine if a child secret were deleted while it was being processed for RPC.
- Fixed an issue preventing a scheduled password change if the secret template expiration is disabled.
- Fixed an issue producing an error during a distributed engine install when the hostname was more than 50 characters.
- Fixed an issue for the ODBC password changer with verification opening the connection but not executing custom commands. The fix allows for: CheckFor, CheckContains, and CheckNotContains to be used in custom commands. Additional logging has also been added when verbose logging is enabled.
- Fixed an issue with RPC on scheduled tasks that have a secret ID on a different trusted domain.
- Fixed an issue where the distributed engine automatic upgrade would fail if the OS account HKU{GUID} entry does not have permission to stop the distributed engine service.

Installation, Upgrade, and Uninstall

- Fixed an issue where ThycoticSetup.exe failed to install SQL Express when selected for the installation.
- Fixed an issue producing an error during a distributed engine install when the hostname was more than 50 characters.

Launchers

Fixed an issue with SAP launcher where record "Additional Processes" stops recording if it was opened from another process within "Process Arguments" and that process is closed.

Reports

Fixed an issue where duplicate report emails were received when a report schedule was not removed after a report with the same name was deleted. The fix does not allow two active reports to have the same report name when creating or updating a report.

Secret Server Cloud

- Fixed an issue where Secret Server Cloud users were presented with a misleading tooltip for user synchronization interval options in directory services configuration. An improved tooltip has been provided to inform the user the limitations of the configuration more accurately in Secret Server Cloud.
- Fixed an issue in Secret Server Cloud where personal folders were not being created for new users created through Delinea One.
- Fixed an issue for Secret Server Cloud where after entering the PIN for Duo and attempting login, Secret Server would activate the Duo Push button in the UI, rather than proceeding with the login.

Secrets, Policies, and Templates

- Fixed an issue by adding more robust permission checks to determine when template conversion options should be available to users, rather than displaying generic access denied errors when users attempted making conversions on a template without the right user roles.
- Fixed an issue by providing additional detail in a warning message to alert the users when duplicating templates that changes must be applied to all secrets intended to use the new template.
- Fixed an issue when setting the auto change schedule on a secret policy to monthly that caused a UI issue that prevented viewing the Remote Password Changing tab.
- Fixed an issue that prevented record retrieval on secret permissions if the application account has owner permissions at the secret level. Now, if you have view access to the secret in the filter you can retrieve the permissions. Queries that do not have secret ID still require the secret owner.
- Fixed an issue where a secret template with a duplicate name in fields would cause a failure when exported, due to the field name match. We recommend using an available slug on the XML for handling fields which have the same name.
- Fixed an issue in the UI when searching of the secrets grid where the search text box covered the first result in the grid.

- Fixed and improved reference labeling to secrets when a secret template changes on deleted secrets. A (deleted) reference is added to secrets used in a secret policy that have been deleted. Added a note to let the user know that deleted secrets are included in the secret count.
- Fixed an issue preventing a scheduled password change if the secret template expiration is disabled.
- Fixed an issue preventing applying a secret policy without access to the required privileged account. Users can now assign a new role that allows users with "list" permissions on the secret to pass the secret access permissions check for privileged and associated secrets when the privileged or associated secrets are enforced by secret policy.

Session Recording

- Fixed an issue causing Session Recording failures to record multiple sessions to the same target with MobaXTerm.
- Fixed an issue where older session recording keystroke logs were not deleted after the configured retention period. Session recording using SSH or RDP Proxy to capture keystroke metadata was not running the cleanup job after the data retention period expired.
- Fixed an issue where searching for recorded sessions could produce an execution timeout error.
- Fixed an issue with the SAP launcher only recording part of the screen.

SSH Proxy and Terminal

- Fixed an issue where SSH proxy sessions via SSH terminal would close within approximately five minutes after launch.
- Fixed an issue where when a distributed engine was used as a proxy server the SSH terminal gave an invalid error during authentication for public keys.
- Fixed an issue when launching SSH secrets via SSH terminal that have 'check out' and 'change on check in' enabled. When exiting the launched session, the secret does not check in.
- Fixed an issue with proxied SSH custom launchers using a "Connect As" secret failing to launch successfully.
- Fixed an issue using private ECDSA SSH keys generated in the OpenSSH format that produced an error when uploading the key on a SSH key template.
- Fixed an issue that caused premature closure of proxied SSH processes when opening multiple sessions of MobaXterm.
- Fixed an issue where the SSH proxy tunnel would not take $password from Xshell client on a custom launcher.
- Fixed an issue where the supported SSH key exchange algorithms would fail to negotiate (ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521).
- Fixed an issue where the keyboard did not operate for users on SSH RDP PuTTY sessions via proxy.

Users and Groups

Fixed an issue with the "Users Migrate to AD" function producing an invalid error when canceling.

Web Password Filler

Fixed an issue where Web Password Filler did not prompt token generation for users on Windows 10 version 20H2.

Fixes made since Early Adopter Version 11.0.000000

- Fixed an issue with secret access requests not allowing the administrator to approve via the inbox.
- Fixed an issue where SSH blocklist caused user keystrokes to be echoed back in the session.
- Fixed an issue with SSH blocklist that caused all su sessions to quit when issuing an exit command.
- Fixed an issue with SSH blocklist where the control-c input does not quit running the program when SSH blocklist is enabled.
- Fixed an issue with SSH blocklist where ESC input on a session with SSH blocklist terminates the session.
- Fixed an issue with SSH blocklist where all su sessions terminate when issuing an exit command.
- Fixed an issue with the lists feature where a user with own or edit permission on a secret were not given access to add or remove lists.
- Fixed date parsing logic for license dates. License dates are in U.S. date format, but default parsing logic uses local-server time-format settings to parse the date and could fail if the expected format is in a different order, for example, in the U.K. where the format is dd/mm/yyyy instead of mm/dd/yyyy. We forced the parser to use the U.S. date format.

Pending Deprecations

This section describes planned feature or platform-support deprecations in Secret Server.

- Internet Explorer 11. Support for Internet Explorer 11 ends on 31 August 2021. Secret Server releases after that date will not support Internet Explorer 11.
- Secret Server Classic UI. The Classic UI option in Secret Server is scheduled to be removed in Q1 2022. After that time, the New UI will be the only available UI option in Secret Server.

Secret Server: 11.0.000005 Release Notes

Release date:

- July 21, 2021 (early release, on-premises only)
- August 17, 2021 (on-premises version)
- August 28, 2021 (Cloud: CA, SG, AU, EU)
- September 11, 2021(Cloud: US)

Note: These dates are tentative. Please see the main release note page for the actually dates as they happen.

Important: If you are in a Secret Server Cloud region that has not yet received the 11.0.x release, please use the "Secret Server: 10.9.000064 Release Notes" on page 1788 notes instead.

Important: If you installed Secret Server as your default or top-level website and you have Privilege Manager (PM) and Secret Server installed together, you may experience the following issues after upgrading to .NET Framework 4.8:

- PM agents will not register.
- When a PM agent updates itself (using the agent utility), it states that there are zero policies to download.

If you believe this scenario applies to you, please contact Delinea Support **before** performing a .NET, Secret Server, or PM upgrade.

Known Issues

Connection Manager Version 1.6.2 Required for Customers Using Secret Server 11.0

Details:

- Intermittent Issue: On a macOS, when run in protocol handler mode, copy and paste does not work. The app does not crash but the copy and paste action is not successful. If the app is “unlocked,” copy, cut, and paste work. For your Secret Server connection to “unlock” it, expand the left navigation panel, and double click on the “lock” icon.
- Intermittent Issue On a macOS, while connected to any Secret Server, if the user edits the same Secret Server name (via the Secret Server connections menu), the app could get stuck in a “connecting…” state.
- Intermittent Issue: On a macOS, some users may experience issues connecting to Secret Server via a Web login (SAML) where the connection stays in a “connecting…” state.
- Upgrade Issues: In some situations, the automatic upgrade may not run the installer after downloading the MSI or package files. On macOS, when upgrading from v1.6.0, the automatic update fails to launch the installer. On Windows, the auto update may not launch the installer. In that case, please download and run the installer manually or via a software deployment tool.

  Installer download links:

  - macOS
  - Windows

New Features and Enhancements

API

Secrets are now searchable by the full path. This removes the need to know the ID to locate the intended secret.

Encrypted Secret Export

Secret Server now allows you to schedule encrypted exports of secret data to external storage.

General

- Updated ESXi integration support via an update of PowerCLI 12.
- Updated PuTTY to version .074.
- Added SNC support for communication between SAP and Secret Server for heartbeat and remote password changing.
- Added support for metadata filters on event pipelines.
- Added the ability to intercept a sudo or su command in a proxied session and inject the session password directly from Secret Server, not disclosing the password to the user.
- Added support for displaying both username and display name in audit logs.

- Added SMB fallback support for local Windows account heartbeats.
- Added a discovery threshold configuration that controls when a dependency is deactivated if it is not found in a scan. The threshold is set to the times a dependency can be missing prior to deactivation. It can also be set to "never" to prevent deactivation.

Inbox

The inbox now provides a customizable toolset to manage how email and notifications are sent and received by users. Inbox allows for configuration of notification scheduling, collecting notifications into digests, creation of message templates and rules, and more.

Lists

Secrets now support configurable data lists. Users with the "apply lists to secrets via secret policies" permission can create a list. This provides an easy mechanism for secret owners to simply choose from provided lists, such as a list of machines the user can select. Additionally, lists can be used for allow and blocklists allowing for control over what the secret owner can access.

Proxy Generic Connections

Secret Server can now securely tunnel a connection to servers operating on a variety of protocols.

Secret Erase

Users with the permissions to use the secret erase feature can permanently erase data from a secret. This provides Secret Server a method to purge secret data without reconciliation of the erased data. The existing "delete" function (now called "deactivate") allows you to "undelete" (reactivate). Secret erase can be audited as an event.

SSH Key Discovery

Secret Server discovery can now discover SSH public keys by scanning key locations on Linux and Unix servers.

User Interface

- Added dependency status reports. The reports are:
  - Overview: Shows how many dependencies failed, succeeded, and were not run.
  - Status: Shows how many dependencies failed, succeeded, and were not run by clickable secret, secret dependency group, and site.
  - Failed by secret: Shows doughnut graphic with secret and fail count.
  - Not run by secret: Shows the not run count by secret.
- Renamed the "Viewing Password Requires Edit" setting to "Hide Launcher Password" in the secret policies editor to improve clarity.

Bug Fixes

Note: The same line item may appear in more than one section when it applies to both.

Access Requests and Secret Workflows

- Fixed an issue where the "Revoke" button was being displayed in the UI after a request had already expired. The button should not be available to users past expiration.
- Fixed an issue where a "No Permission" page is displayed incorrectly when checking-in a secret with "change on check-in" enabled.

Alerts, Events, and Logging

- Fixed an issue where if the browser is set to a non-English language, a failed login attempt causes invalid system logging. The fix changes the character type from text to nvchar to produce successful logging entries. For this data type conversion, we recommend that customers with lots of data may want to run the SQL delta ahead of an upgrade to prevent potential SQL timeouts during upgrade.

API

- Fixed an issue where API was returning null for createDate on SecretSummary queries.
- Fixed an issue that resulted in receiving a 500 error when using the REST API to enable check-in on a secret using the PATCH method /secrets/{id}/security-checkout.
- Fixed an issue where the REST API was returning "Object reference not set to an instance of an object." against the /directory-services/domains/{domainid}/group.
- Fixed an issue for the API endpoint, workflow, or template throwing an exception when the take parameter was not specified.
- Fixed a POST issue in secret-templates REST endpoint that did not properly validate the editablePermission property on fields.
- Fixed an issue where a secret policy which enforces 'View Password Requires Edit' prevents API access to the view the password.

Authentication, Login, and Directory Services

- Fixed an issue with post authentication for SAML.
- Fixed an issue where if the browser is set to a non-English language, a failed login attempt causes invalid system logging. The fix changes the character type from text to nvchar to produce successful logging entries. For this data type conversion, we recommend that customers with lots of data may want to run the SQL delta ahead of an upgrade to prevent potential SQL timeouts during upgrade.
- Fixed an issue where when a distributed engine was used as a proxy server the SSH terminal gave an invalid error during authentication for public keys.
- Fixed an issue where occasionally users that synced via Active Directory were not getting assigned default role (s).
- Fixed an issue with Active Directory synchronization throwing a credential validation error when the sync credentials are from another domain after creating relationship between domains.
- Added a notice in the UI for directory services group sync search results being limited to 1000 groups.

- Fixed an issue where Active Directory group names were showing with their pre-Windows 2000 name instead of the group name.
- Fixed an issue where Secret Server Cloud users were presented with a misleading tooltip for user synchronization interval options in directory services configuration. An improved tooltip has been provided to inform the user the limitations of the configuration more accurately in Secret Server Cloud.

Discovery

- Fixed an issue in discovery when using LDAPS where discovery attempts to use port 389 despite the user having selected LDAPS port 636.
- Fixed an issue with saving IP addresses additions to the manual host range within the discovery scanner. A message is now provided to inform the user that the manual host ranges should be set on the discovery source scanner, rather than as a default setting on the scanner itself.
- Fixed an issue that could occasionally cause dependencies to get disabled by a computer discovery scan.
- Fixed a discovery issue that could cause background processes to stop.
- Fixed an issue where discovery was retaining too many historical records for each host. We added an advanced configuration setting to define the number of records.
- Fixed a discovery issue where when the discoverysourceid was equal to 9 or 10, after the domain scan completed to get machines, the Service Account tab in the UI would not load re-scan buttons.
- Fixed an issue where the schedule task scanner did not work for the domain discovery source with two UPNs.
- Fixed an issue with discovery for scheduled task scanning across multiple trusted domains.
- Fixed a discovery issue where similar OU names resulted in discovery importing accounts from an OU with a similar name to the targeted OU.

Event Subscriptions and Pipelines

- Fixed an issue that could be encountered with event subscriptions after converting a secret's template. An improved message is now provided in the UI when converting secret templates to inform the user what actions needs to take place to prevent the issue.
- Fixed an issue with event pipelines receiving an application error when trying to add tasks in a policy if the policy has a deactivated pipeline.
- Fixed an event subscriptions issue of not initiating for engine events.

Folders

- Fixed an issue when editing folder permissions and receiving no search results for groups that contained a '+' in the group name.
- Fixed an issue that prevented the ability to sort folders by alphabetical order on name in the Favorites tab.
- Fixed an issue where the folder name may not display in the secret grid.
- Fixed an issue where right clicking secret folders in the UI did not have the "View Details" option available for users with view permissions.

- Fixed an issue where editing folder sharing permissions removed restricted template options in the new UI.
- Fixed an issue that caused an application error when changing a folder's secret permissions to "none."

General

- Fixed an issue that caused a 500 error if certain password sequences were used when password validation settings were enabled to prevent password sequences.
- Fixed an issue that prevented content deletion in metadata fields.
- Fixed an issue where pressing enter on the keyboard when using the main application search box did not display results in the secret grid but only in the dropdown.
- Fixed an issue where the "Copy to Clipboard" button was not appearing on fields that contained a character count beyond the max field character display.
- Fixed an issue where the "Options" menu on pages is automatically hidden in the UI if no options are present to display to the user.
- Fixed an issue where sessions created through the Secret Server terminal "Launch" command were not getting returned by session search when filtering by secret name, secret field value, or user.
- Fixed an issue where on the home page engine status widget was not displaying the correct status in the "Connection Status 1" column. In the fix, this column has been renamed to 'Activation Status'.
- Fixed an invalid audit message entry when a user initiates "Change Password Now" using the "Randomly Generated" option.
- Fixed an issue where changing the name of a launcher did not update the name of the launcher on the secret.
- Fixed a mouseover label on "RADIUS User Name" on the Users page. Mouseover tooltip now displays "The user name of your RADIUS user."
- Fixed an issue with password requirement validation. When a password policy specified that the username must not be included in the password, the validation was too aggressive. Now at least three consecutive characters in the password must match a section of the username for a password to be rejected.
- Fixed an issue that caused locks to hang in tbDatabaseCache.
- Fixed an issue where an account is not being used as a "Privileged Secrets for Scripts" when importing for discovery for RPC and heartbeat.
- Fixed an issue where RDP proxy may experience intermittent connectivity when using the Mac version of Connection Manager.
- Fixed an issue causing Secret Server-BWSR errors from UDP Syslog datagrams exceeding RFC3164/5424 specifications. Note that TCP or SecureTCP, is preferred over UDP for reliability purposes.
- Fixed an issue when editing the default password requirement when using a language other than English. It set the default to false.
- Fixed an issue when syncing AzureAD where deleting a sync group in AzureAD would result in members of all groups being synchronized.

Heartbeat, Distributed Engines, and RPC

- Fixed an issue that resulted in a "Phantom.JS.exe not found" error when trying to run a heartbeat on a Web password with a distributed engine.
- Fixed an issue where password changing for secrets was not being triggered after a forced check-in.
- Fixed an issue when creating custom dependencies that produced an invalid error when adding a parent field of 'Machine.'
- Fixed an issue when setting the auto change schedule on a secret policy to monthly that caused a UI issue that prevented viewing the Remote Password Changing tab.
- Fixed an issue where a "Last Heartbeat Status" error was displayed in the UI that indicated the user's Web browser stopped polling the server. An improved error message is provided in cases where polling timeout still occurs.
- Fixed an issue with password changing for SAP secrets that caused failures at password change due to an issue with the check-in.

  For SAP servers where rfc/reject_expired_passwd = 1, a new option was added to the Advanced Settings in the SAP password changer. This new option, "Use Single Destination (SAP)" is false by default, but, when set to true, it allows privileged password changing to succeed on these servers by using the privileged credentials in both steps of the privileged password change.

  For servers where rfc/reject_expired_passwd = 0, this option may be set to true or false and password changing will succeed.
- Fixed an issue where previously inactive custom scripts were still displayed as available for RPC.
- Fixed an issue where a secret policy which enforces 'View Password Requires Edit' prevents API access to the view the password.
- Fixed an issue that displayed the distributed engine count incorrectly based on previously deleted distributed engines.
- Fixed an issue that could prevent the privileged account password from changing by a distributed engine if a child secret were deleted while it was being processed for RPC.
- Fixed an issue preventing a scheduled password change if the secret template expiration is disabled.
- Fixed an issue producing an error during a distributed engine install when the hostname was more than 50 characters.
- Fixed an issue for the ODBC password changer with verification opening the connection but not executing custom commands. The fix allows for: CheckFor, CheckContains, and CheckNotContains to be used in custom commands. Additional logging has also been added when verbose logging is enabled.
- Fixed an issue with RPC on scheduled tasks that have a secret ID on a different trusted domain.
- Fixed an issue where the distributed engine automatic upgrade would fail if the OS account HKU{GUID} entry does not have permission to stop the distributed engine service.

Installation, Upgrade, and Uninstall

- Fixed an issue where ThycoticSetup.exe failed to install SQL Express when selected for the installation.
- Fixed an issue producing an error during a distributed engine install when the hostname was more than 50 characters.

Launchers

Fixed an issue with SAP launcher where record "Additional Processes" stops recording if it was opened from another process within "Process Arguments" and that process is closed.

Reports

Fixed an issue where duplicate report emails were received when a report schedule was not removed after a report with the same name was deleted. The fix does not allow two active reports to have the same report name when creating or updating a report.

Secret Server Cloud

- Fixed an issue where Secret Server Cloud users were presented with a misleading tooltip for user synchronization interval options in directory services configuration. An improved tooltip has been provided to inform the user the limitations of the configuration more accurately in Secret Server Cloud.
- Fixed an issue in Secret Server Cloud where personal folders were not being created for new users created through Delinea One.
- Fixed an issue for Secret Server Cloud where after entering the PIN for Duo and attempting login, Secret Server would activate the Duo Push button in the UI, rather than proceeding with the login.

Secrets, Policies, and Templates

- Fixed an issue by adding more robust permission checks to determine when template conversion options should be available to users, rather than displaying generic access denied errors when users attempted making conversions on a template without the right user roles.
- Fixed an issue by providing additional detail in a warning message to alert the users when duplicating templates that changes must be applied to all secrets intended to use the new template.
- Fixed an issue when setting the auto change schedule on a secret policy to monthly that caused a UI issue that prevented viewing the Remote Password Changing tab.
- Fixed an issue that prevented record retrieval on secret permissions if the application account has owner permissions at the secret level. Now, if you have view access to the secret in the filter you can retrieve the permissions. Queries that do not have secret ID still require the secret owner.
- Fixed an issue where a secret template with a duplicate name in fields would cause a failure when exported, due to the field name match. We recommend using an available slug on the XML for handling fields which have the same name.
- Fixed an issue in the UI when searching of the secrets grid where the search text box covered the first result in the grid.

- Fixed and improved reference labeling to secrets when a secret template changes on deleted secrets. A (deleted) reference is added to secrets used in a secret policy that have been deleted. Added a note to let the user know that deleted secrets are included in the secret count.
- Fixed an issue preventing a scheduled password change if the secret template expiration is disabled.
- Fixed an issue preventing applying a secret policy without access to the required privileged account. Users can now assign a new role that allows users with "list" permissions on the secret to pass the secret access permissions check for privileged and associated secrets when the privileged or associated secrets are enforced by secret policy.

Session Recording

- Fixed an issue causing Session Recording failures to record multiple sessions to the same target with MobaXTerm.
- Fixed an issue where older session recording keystroke logs were not deleted after the configured retention period. Session recording using SSH or RDP Proxy to capture keystroke metadata was not running the cleanup job after the data retention period expired.
- Fixed an issue where searching for recorded sessions could produce an execution timeout error.
- Fixed an issue with the SAP launcher only recording part of the screen.

SSH Proxy and Terminal

- Fixed an issue where SSH proxy sessions via SSH terminal would close within approximately five minutes after launch.
- Fixed an issue where when a distributed engine was used as a proxy server the SSH terminal gave an invalid error during authentication for public keys.
- Fixed an issue when launching SSH secrets via SSH terminal that have 'check out' and 'change on check in' enabled. When exiting the launched session, the secret does not check in.
- Fixed an issue with proxied SSH custom launchers using a "Connect As" secret failing to launch successfully.
- Fixed an issue using private ECDSA SSH keys generated in the OpenSSH format that produced an error when uploading the key on a SSH key template.
- Fixed an issue that caused premature closure of proxied SSH processes when opening multiple sessions of MobaXterm.
- Fixed an issue where the SSH proxy tunnel would not take $password from Xshell client on a custom launcher.
- Fixed an issue where the supported SSH key exchange algorithms would fail to negotiate (ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521).
- Fixed an issue where the keyboard did not operate for users on SSH RDP PuTTY sessions via proxy.

Users and Groups

Fixed an issue with the "Users Migrate to AD" function producing an invalid error when canceling.

Web Password Filler

Fixed an issue where Web Password Filler did not prompt token generation for users on Windows 10 version 20H2.

Fixes made since Early Adopter Version 11.0.000000

- Fixed an issue with secret access requests not allowing the administrator to approve via the inbox.
- Fixed an issue where SSH blocklist caused user keystrokes to be echoed back in the session.
- Fixed an issue with SSH blocklist that caused all su sessions to quit when issuing an exit command.
- Fixed an issue with SSH blocklist where the control-c input does not quit running the program when SSH blocklist is enabled.
- Fixed an issue with SSH blocklist where ESC input on a session with SSH blocklist terminates the session.
- Fixed an issue with SSH blocklist where all su sessions terminate when issuing an exit command.
- Fixed an issue with the lists feature where a user with own or edit permission on a secret were not given access to add or remove lists.

Pending Deprecations

This section describes planned feature or platform-support deprecations in Secret Server.

- Internet Explorer 11. Support for Internet Explorer 11 ends on 31 August 2021. Secret Server releases after that date will not support Internet Explorer 11.
- Secret Server Classic UI. The Classic UI option in Secret Server is scheduled to be removed in Q1 2022. After that time, the New UI will be the only available UI option in Secret Server.

Secret Server: 10.9.000064 Release Notes

10.9.000065 is a retroactive patch for this release. Please see "Secret Server 11.7.000001 Release Notes" on page 1596 for details.

Release date:

- April 13, 2021 (on-premises version)
- April 3 - May 15th, 2021. Release date is dependent on region (cloud version)

Important: These notes cover the General Availability release of version 10.9.000064. The general release is not till April 13, 2021 for the on-premises version and between April 3rd and May 15th 2021, depending on region, for the cloud version. If you are not part of the early release program, or are in a Secret Server Cloud region that has not yet received the 10.9.00064 release, please use the "Secret Server: 10.9.000005/33 Release Notes" on page 1798 notes instead.

Updates:

- April 3rd: 10.9.000064 released to Secret Server Cloud in the Canada and Southeast Asia regions. Secret Server
- April 10th: 10.9.000064 released to Secret Server Cloud in the Australia region.
- April 13th: 10.9.000064 On-Premises is released.
- April 24th: 10.9.000064 released to half of the United States region customers.

Important: If you installed Secret Server as your default or top-level website and you have Privilege Manager (PM) and Secret Server installed together, you may experience the following issues after upgrading to .NET Framework 4.8:

- PM agents will not register.
- When a PM agent updates itself (using the agent utility), it states that there are zero policies to download.

If you believe this scenario applies to you, please contact Delinea Support **before** performing a .NET, Secret Server, or PM upgrade.

Upgrade Notes

Delinea encourages all customers to upgrade at the earliest opportunity.

New Features and Enhancements

SSH Command Blocklisting

SSH command blocklisting allows administrators to configure lists of commands that a user may not enter during a proxied SSH session.

Pre-Check-in Pipeline Trigger

An additional trigger condition is available for secret event policy pipelines. Pre-check-in triggers apply when a secret check in is requested, before the check in occurs. The trigger can stop the pipeline with an error message or run commands.

Configuration Management

Configuration management adds export and import of Secret Server configuration settings.

This facilitates transferring configurations from test to production environments. It also makes setup and configuration of new Secret Server instances easier for customers and partners with multiple installations.

Timeouts for Tiered Workflow

Workflow templates have new timeout configuration options. They allow a workflow to automatically advance to the next step if insufficient responses are received during a configurable period.

Selectable Workflow Steps

Workflow approvals now allow an administrator to choose the next action when a step of the workflow is approved. Available actions are:

- Approving the entire workflow, disregarding any later steps
- Moving to the next step of the workflow
- Moving to a later step in the workflow

CSV Scheduled Reports

Scheduled report configuration now includes the option to attach the report to the scheduled email in a spreadsheet-friendly comma-separated value (CSV) file.

User Interface

- Migrated to an Angular framework which better supports usability and accessibility.
- Left navigation menu is now a menu role, which helps accessibility and quickens keyboard navigation.
- Updated color palette for accessibility.
- Updated numerous buttons and links that were previously inaccessible with keyboard to improve keyboard navigation and accessibility.
- Fixed an issue that caused selection and drop-down components to clip behind scrollable elements.
- Improved labeling and tool tips for Connection Manager configuration.
- Added a column in the new UI for admins in the users grid for 2FA. The same was added to API user search results.
- The secret detail side menu now stays present in the UI after navigating away from secret.

General

- Added user-defined metadata sections and field values on users, groups, folders, or secrets.
- Updated and expanded the "custom launcher arguments" field character limit to prevent impact on scripted launchers.
- Added a redirect service to direct Secret Server internal hyperlink traffic to the updated Delinea documentation portal.
- Clarified the heartbeat status message on the secret page.
- Added additional license key support for the Privilege Manager Unix/Linux Server.
- Improved user data entry input security during Connect As sessions.

Bug Fixes

API

- Fixed an issue when creating workflow steps due to the template step stub not being available for the configuration object.
- Fixed an issue returning a 500 error when getting the triggers for an event pipeline (`/api/v1/event-pipeline/{id}/trigger`).
- Corrected the documentation on the endpoint for OAuth to obtain a refresh token.
- Corrected the documentation to properly cite that a description is required for ReportCreateArgs to create reports.

Heartbeat and RPC

- Fixed an issue with heartbeat on an AD Sync that prompted an exception.
- Fixed an issue where the test action of SSH RPC configuration could not handle a dollar sign as the first character of the password.
- Fixed an issue with a RPC and heartbeat PowerShell script that failed when a privileged account lacked additional permissions.
- Fixed an issue that caused RPC to have "invalid" failures if values in the password changer were blank, even though the fields were not mandatory according to the configuration.

SSH Terminal

- Fixed an issue where the SSH terminal did not honor host restrictions for secret launches.
- Fixed an issue where SSH command menu commands through the SSH terminal did not reset the SSH proxy's timer, causing an inactivity timeout.
- Fixed an issue where the UI component for the SSH terminal blocklist settings was not visible when the SSH terminal setting was set to "No."
- Fixed an issue that locked Secret Server user accounts when Unix connections to SSH terminal were made with a keypair and no passphrase. Authentication prompts are determined by Secret Server > Admin > Config > Login > Enable SSH Key Integration (for SSH terminal). If this check box is left unchecked (assuming SSH proxy and terminal are enabled), the user is prompted for a password after trying to SSH into Secret Server. If the box is checked, no attempts to log on with a private key are allowed if "Password Only" is selected. If one of the three other Unix authentication methods are selected, the private key is attempted first (or exclusively, in the case of "Public Key Only").

User Interface

- Fixed an issue where the UI would freeze when attempting to edit the manual host range on the discovery scanners page.
- Fixed an issue where in the UI it did not properly display a site after being disabled from a discovery source on a domain.
- Fixed an issue where the UI would not automatically adjust to display scrolling in modal windows.
- Fixed an issue in the new UI where moving a folder would not correctly inherit the new parent folder's permissions as displayed.
- Fixed an issue in the new UI where a validation error was not displayed when a folder was moved to a folder containing a folder of the same name. The fix provides an error message to notify the user and allows the user to pick a new destination.
- Fixed an issue in the new UI where a group search was not restricted by owner.
- Fixed an issue with the RPC's associated secrets table hiding columns after a column was resized beyond the workspace.

General

- Fixed an issue impacting proper display of last login record in audits for OpenID Connect.
- Fixed an issue that could cause discovery logs to take longer than usual to load and display.
- Fixed an issue in Active Directory domain settings that caused the synchronization secret to revert to a previous entry.
- Fixed an issue that prevented using some identifiers on the distinguished name in directory services. Added a base distinguished name code in the LDAP field configuration to ensure some identifiers do not cause an error.
- Fixed an issue for Okta SAML login failures due to accounts being re-enabled.
- Fixed an issue for RDP proxying where the remote host name was not being passed to the protocol handler for the RDP client to display.
- Fixed an issue where additional DLL files were removed when a distributed engine was upgraded. A data directory "ignore list" file was added to list files that shall not be deleted during the upgrade process.
- Fixed an issue that prevented a secret's security settings from being edited when an event pipeline policy was set through a secret policy.
- Fixed an issue that prompted a file upload error on a folder when attempting to attach a file. This occurred when the folder included a secret policy with an event pipeline policy.
- Fixed an issue with protocol handler where an encoder was initializing even if session recording was disabled. Now the encoder is loaded only if screenshots are taken.
- Fixed an issue with the C# SDK returning a null after the cache has expired for CacheThenServerAllowExpired.
- Fixed an issue that could prevent users from logging in using SAML.
- Fixed an issue in the new UI that prevented domain selection when creating a new user.
- Fixed an issue where the site connector for a local site would not accept changes to the configuration in the UI. This only applied to system sites.
- Fixed a permissions issue that occurred when enabling "view requires edit" on a password field that caused a user with only view permissions on a secret to no longer view the one-time password.
- Fixed an issue with the SAML log on workflow that caused an invalid login page redirection loop with OpenID Connect.
- Fixed an issue when using the move folder command causing removal of secret policy settings and inherit permissions.
- Fixed an issue with SDK client management in client onboarding improperly displaying disabled application accounts when editing a user.
- Fixed an issue with session recording the caused inactivity timeouts when activity timeout was not enabled.
- Fixed an issue that caused SMTP send failures when the mail server was configured for implicit SSL.
- Fixed an issue affecting 2FA login when OATH was not properly configured for the user. Log on now redirects to an error page with indication of the configuration issue.
- Fixed an issue when disabled users were removed from local groups when new members were added to the group.

- Fixed an issue where browsers can become unresponsive when producing long-running reports.
- Fixed an issue where Secret Server was not sending metadata to PBA when assigned to a site with a distributed engine, preventing SSO.
- Fixed an issue where a permissions error is incorrectly prompted when setting a favorite folder.
- Fixed an issue with SecureBlackbox TISSHClient not being compatible with -etm MAC algorithms. An update for SecureBlackbox has resolved this.
- Fixed an issue where in SSC where the SSH proxy port was not being accepted.
- Fixed an issue in 10.9 Upgrade where SQL was missing the DBO schema when creating the table tbEngineServerCapabilties, producing an "object not found" error when verifying the deltas.
- Fixed an intermittent issue causing an error when using RDP proxy session recording with keystroke recording enabled. The error was intermittent based on which engine handled the connection.
- Fixed an issue where custom logos set in Secret Server did not display correctly caused by browser-level caching.
- Fixed an issue where secrets were not checking in after the launcher is closed while using RDP proxy.
- Fixed an issue where setting the "Allow Duplicate Secret Names" permission option to "No" caused a bulk delete action to produce an error.

Fixes made since Early Adopter Version 10.9.000063

- Fixed an issue where execution times were longer than expected (waiting for a full timeout) for SSH scripts via the password changers (heartbeat and RPC), discovery, or checkin and checkout hooks.
- Fixed an issue in the Early Adopter 10.9.000063 version where possible memory leaks could occur.

Pending Deprecations

This section describes planned feature or platform-support deprecations in Secret Server.

- Internet Explorer 11: Support for Internet Explorer 11 will end on 31 August 2021. Secret Server releases after that date will not support Internet Explorer 11.
- Secret Server Classic UI: The Classic UI option in Secret Server is scheduled for removal in Q1 2022. After the announced date, the New UI will be the only on in Secret Server.

Secret Server: 10.9.000063 Release Notes

Release date:

- March 23, 2021 (On-Premise Early Adopter)
- April 13, 2021 (On-Premises)
- April 3 to May 15, 2021 (Cloud–depends on region)

Important: These notes cover the Early Adopter version 10.9.000063. The general availability release is not till April 13, 2021 for the on-premises version and between April 3rd and May 15th 2021, depending on region, for the cloud version. If you are not part of the early release program, please use the "Secret Server: 10.9.000005/33 Release Notes" on page 1798 notes instead.

Important: If you installed Secret Server as your default or top-level website and you have Privilege Manager (PM) and Secret Server installed together, you may experience the following issues after upgrading to .NET Framework 4.8:

- PM agents will not register.
- When a PM agent updates itself (using the agent utility), it states that there are zero policies to download.

If you believe this scenario applies to you, please contact Delinea Support **before** performing a .NET, Secret Server, or PM upgrade.

Upgrade Notes

Delinea encourages all customers to upgrade at the earliest opportunity.

New Features and Enhancements

SSH Command Blocklisting

SSH command blocklisting allows administrators to configure lists of commands that a user may not enter during a proxied SSH session.

Pre-Check-in Pipeline Trigger

An additional trigger condition is available for secret event policy pipelines. Pre-check-in triggers apply when a secret check in is requested, before the check in occurs. The trigger can stop the pipeline with an error message or run commands.

Configuration Management

Configuration management adds export and import of Secret Server configuration settings.

This facilitates transferring configurations from test to production environments. It also makes setup and configuration of new Secret Server instances easier for customers and partners with multiple installations.

Timeouts for Tiered Workflow

Workflow templates have new timeout configuration options. They allow a workflow to automatically advance to the next step if insufficient responses are received during a configurable period.

Selectable Workflow Steps

Workflow approvals now allow an administrator to choose the next action when a step of the workflow is approved. Available actions are:

- Approving the entire workflow, disregarding any later steps
- Moving to the next step of the workflow
- Moving to a later step in the workflow

CSV Scheduled Reports

Scheduled report configuration now includes the option to attach the report to the scheduled email in a spreadsheet-friendly comma-separated value (CSV) file.

User Interface

- Migrated to an Angular framework which better supports usability and accessibility.
- Left navigation menu is now a menu role, which helps accessibility and quickens keyboard navigation.
- Updated color palette for accessibility.
- Updated numerous buttons and links that were previously inaccessible with keyboard to improve keyboard navigation and accessibility.
- Fixed an issue that caused selection and drop-down components to clip behind scrollable elements.
- Improved labeling and tool tips for Connection Manager configuration.
- Added a column in the new UI for admins in the users grid for 2FA. The same was added to API user search results.
- The secret detail side menu now stays present in the UI after navigating away from secret.

General

- Added user-defined metadata sections and field values on users, groups, folders, or secrets.
- Updated and expanded the "custom launcher arguments" field character limit to prevent impact on scripted launchers.
- Added a redirect service to direct Secret Server internal hyperlink traffic to the updated Delinea documentation portal.
- Clarified the heartbeat status message on the secret page.
- Added additional license key support for the Privilege Manager Unix/Linux Server.
- Improved user data entry input security during Connect As sessions.

Bug Fixes

API

- Fixed an issue when creating workflow steps due to the template step stub not being available for the configuration object.
- Fixed an issue returning a 500 error when getting the triggers for an event pipeline (`/api/v1/event-pipeline/{id}/trigger`).
- Corrected the documentation on the endpoint for OAuth to obtain a refresh token.
- Corrected the documentation to properly cite that a description is required for ReportCreateArgs to create reports.

Heartbeat and RPC

- Fixed an issue with heartbeat on an AD Sync that prompted an exception.
- Fixed an issue where the test action of SSH RPC configuration could not handle a dollar sign as the first character of the password.
- Fixed an issue with a RPC and heartbeat PowerShell script that failed when a privileged account lacked additional permissions.
- Fixed an issue that caused RPC to have "invalid" failures if values in the password changer were blank, even though the fields were not mandatory according to the configuration.

SSH Terminal

- Fixed an issue where the SSH terminal did not honor host restrictions for secret launches.
- Fixed an issue where SSH command menu commands through the SSH terminal did not reset the SSH proxy's timer, causing an inactivity timeout.
- Fixed an issue where the UI component for the SSH terminal blocklist settings was not visible when the SSH terminal setting was set to "No."
- Fixed an issue that locked Secret Server user accounts when Unix connections to SSH terminal were made with a keypair and no passphrase. Authentication prompts are determined by Secret Server > Admin > Config > Login > Enable SSH Key Integration (for SSH terminal). If this check box is left unchecked (assuming SSH proxy and terminal are enabled), the user is prompted for a password after trying to SSH into Secret Server. If the box is checked, no attempts to log on with a private key are allowed if "Password Only" is selected. If one of the three other Unix authentication methods are selected, the private key is attempted first (or exclusively, in the case of "Public Key Only").

User Interface

- Fixed an issue where the UI would freeze when attempting to edit the manual host range on the discovery scanners page.
- Fixed an issue where in the UI it did not properly display a site after being disabled from a discovery source on a domain.
- Fixed an issue where the UI would not automatically adjust to display scrolling in modal windows.
- Fixed an issue in the new UI where moving a folder would not correctly inherit the new parent folder's permissions as displayed.
- Fixed an issue in the new UI where a validation error was not displayed when a folder was moved to a folder containing a folder of the same name. The fix provides an error message to notify the user and allows the user to pick a new destination.
- Fixed an issue in the new UI where a group search was not restricted by owner.
- Fixed an issue with the RPC's associated secrets table hiding columns after a column was resized beyond the workspace.

General

- Fixed an issue impacting proper display of last login record in audits for OpenID Connect.
- Fixed an issue that could cause discovery logs to take longer than usual to load and display.
- Fixed an issue in Active Directory domain settings that caused the synchronization secret to revert to a previous entry.
- Fixed an issue that prevented using some identifiers on the distinguished name in directory services. Added a base distinguished name code in the LDAP field configuration to ensure some identifiers do not cause an error.
- Fixed an issue for Okta SAML login failures due to accounts being re-enabled.
- Fixed an issue for RDP proxying where the remote host name was not being passed to the protocol handler for the RDP client to display.
- Fixed an issue where additional DLL files were removed when a distributed engine was upgraded. A data directory "ignore list" file was added to list files that shall not be deleted during the upgrade process.
- Fixed an issue that prevented a secret's security settings from being edited when an event pipeline policy was set through a secret policy.
- Fixed an issue that prompted a file upload error on a folder when attempting to attach a file. This occurred when the folder included a secret policy with an event pipeline policy.
- Fixed an issue with protocol handler where an encoder was initializing even if session recording was disabled. Now the encoder is loaded only if screenshots are taken.
- Fixed an issue with the C# SDK returning a null after the cache has expired for CacheThenServerAllowExpired.
- Fixed an issue that could prevent users from logging in using SAML.
- Fixed an issue in the new UI that prevented domain selection when creating a new user.
- Fixed an issue where the site connector for a local site would not accept changes to the configuration in the UI. This only applied to system sites.
- Fixed a permissions issue that occurred when enabling "view requires edit" on a password field that caused a user with only view permissions on a secret to no longer view the one-time password.
- Fixed an issue with the SAML log on workflow that caused an invalid login page redirection loop with OpenID Connect.
- Fixed an issue when using the move folder command causing removal of secret policy settings and inherit permissions.
- Fixed an issue with SDK client management in client onboarding improperly displaying disabled application accounts when editing a user.
- Fixed an issue with session recording the caused inactivity timeouts when activity timeout was not enabled.
- Fixed an issue that caused SMTP send failures when the mail server was configured for implicit SSL.
- Fixed an issue affecting 2FA login when OATH was not properly configured for the user. Log on now redirects to an error page with indication of the configuration issue.
- Fixed an issue when disabled users were removed from local groups when new members were added to the group.

- Fixed an issue where browsers can become unresponsive when producing long-running reports.
- Fixed an issue where Secret Server was not sending metadata to PBA when assigned to a site with a distributed engine, preventing SSO.
- Fixed an issue where a permissions error is incorrectly prompted when setting a favorite folder.
- Fixed an issue with SecureBlackbox TISSHClient not being compatible with -etm MAC algorithms. An update for SecureBlackbox has resolved this.
- Fixed an issue where in SSC where the SSH proxy port was not being accepted.
- Fixed an issue in 10.9 Upgrade where SQL was missing the DBO schema when creating the table tbEngineServerCapabilties, producing an "object not found" error when verifying the deltas.
- Fixed an intermittent issue causing an error when using RDP proxy session recording with keystroke recording enabled. The error was intermittent based on which engine handled the connection.
- Fixed an issue where custom logos set in Secret Server did not display correctly caused by browser-level caching.
- Fixed an issue where secrets were not checking in after the launcher is closed while using RDP proxy.
- Fixed an issue where setting the "Allow Duplicate Secret Names" permission option to "No" caused a bulk delete action to produce an error.

Pending Deprecations

This section describes planned feature or platform-support deprecations in Secret Server.

- Internet Explorer 11: Support for Internet Explorer 11 will end on 31 August 2021. Secret Server releases after that date will not support Internet Explorer 11.
- Secret Server Classic UI: The Classic UI option in Secret Server is scheduled for removal in Q1 2022. After the announced date, the New UI will be the only on in Secret Server.

Secret Server: 10.9.000005/33 Release Notes

Release date:

- December 14, 2020 (on-premises version)
- December 12, 2020 (cloud version)

Important: If you installed Secret Server as your default or top-level website and you have Privilege Manager (PM) and Secret Server installed together, you may experience the following issues after upgrading to .NET Framework 4.8:

- PM agents will not register.
- When a PM agent updates itself (using the agent utility), it states that there are zero policies to download.

If you believe this scenario applies to you, please contact Delinea Support **before** performing a .NET, Secret Server, or PM upgrade.

Important: This release corrects a critical issue discovered in Secret Server version 10.9.000032. Delinea encourages all customers to upgrade any 10.9.00032 installations to 10.9.000033 as soon as possible. For more

information, including remediation (if you have been affected), see the "Secret Server: 10.9.000005/32 Release Notes" on page 1804.

Note: All the original release notes for 10.9.000032 have been moved here for convenience and clarity.

Upgrade Notes

Delinea encourages all customers to upgrade at the earliest opportunity.

Note: This version of Secret Server uses a two-step upgrade. See Secret Server: 10.9.000005/33 Release Notes.

New Features and Enhancements

HSM Key Rotation

This feature enables customers using a Hardware Security Module (HSM) to rotate their root encryption key. Secret Server supports using an HSM to store the root encryption key for all data in the Secret Server database. Some customers have security policies requiring rotation of all encryption keys. Until now this was a manual process.

DSV Integration: SSH Key Synchronization

Further enhancing the integration between DevOps Secrets Vault (DSV) and Secret Server, the integration now supports customers using DSV to provide PAM services to their DevOps teams. DevOps teams may use SSH keys to access portions of their infrastructure.

SSH Key Authentication to SSH Terminal

SSH terminal in Secret Server provides SSH access for users who do not want to use the Secret Server UI. This is particularly useful for Linux users. These users commonly use SSH keys to authenticate to servers and services. Providing SSH key authentication to SSH terminal allows them to continue to use their preferred access tools with Secret Server.

User Interface

- Secret Server now uses the terms allowlist and denylist. See Delinea shifts the language used in products and materials to promote inclusivity.
- New content has been added to "empty" user interface pages to explain initial actions to be taken by new users to streamline the onboarding experience. There is now more information about what they are able to do with helpful links to the knowledge base and corresponding technical documentation, giving them more confidence as they navigate Secret Server for the first time.
- Ongoing new user interface updates include pages such as User Management, Login, Reports, and Distributed Engine configuration.

General

- Added the ability to trigger event pipelines on the heartbeat UnableToConnect status. This feature adds a new advanced configuration setting, "Heartbeat: Include UnableToConnect as Heartbeat Failure Event". When toggled to true, this setting allows a user to include UnableToConnect as part of the heartbeat failure event subscriptions. It defaults to false.

- Improved handling of site-based requests in relation to heartbeats and remote password changer.
- Secret Server now integrates with Slack, allowing for notifications and workflow handling. This includes approval requests, recently used secret notifications, and launching secrets.

Security

- Added protection against malicious remote code execution through CEF log file tampering.

  Note: Only applies to Secret Server on-premise.
- Added protection against malicious argument injection in SSH launcher.

API

- Fixed an issue to properly display API_AccessDenied as a 403-error code.
- Updated error handling and prompts when editing a group via Active Directory sync. The error for the add has been changed to API_CannotAddRemoveUsersFromDirectoryManagedGroups for clarity. The remove has also been updated to throw that error when the group is from AD sync.
- Created the following new endpoints:
  - Distributed engine
  - Dual controls
  - Report schedule
  - Secret policy search
  - Slack
  - Ticket System
  - User and group management

Integrations

- Fixed an issue with ServiceNow validation that caused tickets in "Active" status to fail validation. Added an active status to the default allowed statues for ServiceNow. This auto-populates when ServiceNow is setup.
- Fixed an issue when creating a ServiceNow ticket without a secret caused a failure. The system now validates the secret field being required before proceeding.

Protocol Handler

Protocol handler is updated for this release. Added Secret Server protocol handler administrative settings that you can configure through Microsoft's Group Policy Objects (GPOs) or through Secret Server itself. One setting controls which domains, URLs, or IP addresses the protocol handler installation may connect to. The other setting ensures protocol handler will never auto-update itself, even if told to by the Secret Server installation that it connects to.

Bug Fixes

- Fixed an issue causing duplication of tbOAuthExpiration. Added process that deletes a tbOAuthExpiration object if one was created earlier in the same request and another one is about to be created.

- Fixed an issue where the date format displays incorrectly according to the date preference set by the user.
- Fixed an issue in RPC where a space in the character set fails PowerShell password changes. We now enclose all parameters that go to the PowerShell Password Changer in double quotes.
- Fixed an issue preventing authentication to VMware with API via PowerShell. All fields of associated secrets are available in RPC scripts now.
- Fixed an issue causing the discovery start time to increase after each interval based on extending completion times. Changed the logic for both discovery and computer scan to run based upon the start date and not run if a scan is in process.
- Fixed an issue with UNIX discovery search filter not matching the machine scanner. Modified a tool tip indicating that $machine filters cannot be used with host range and machine scanners.
- Fixed a script issue for ParseDataItems not handling edge case—System.NullReferenceException: Object reference not set to an instance of an object. Added a null check for the return object in ParseDataItems.
- Fixed an event subscription issue where the engine offline email alert was not being sent.
- Fixed an issue for SSH-proxied secret's audit log where no results are displayed when selecting "Show Proxy Credentials."
- Fixed an issue for RDP-proxied secret's audit log where no results are displayed when selecting "Show Proxy Credentials."
- Fixed an issue that when navigating manually to an enabled secret from a deleted secret erroneously retained the deleted banner.
- Fixed an issue where users are not automatically re-enabled while using "Automatic User Management" with SAML.
- Fixed an issue that caused inability to bulk assign privileged accounts for Unix secrets. Added new "Update Associated Secrets" bulk operation which sets the associated secret list on any selected secrets.
- Fixed an issue where the folder search was not functioning when importing accounts to a folder.
- Fixed an issue causing Oracle scripts to not execute successfully on remote distributed engine.
- Fixed an issue with the left navigation folder tree navigation causing the horizontal scroll to function incorrectly.
- Fixed an issue with Office 365 and Azure AD causing a wrong heartbeat status code and preventing consecutive attempts. Added code to differentiate authentication errors coming back from Office 365 from other errors. Authentication Errors will continue to create a "LoginFailed" error on the result while all other errors will generate "UnableToConnect".
- Fixed an issue where the user is unable to see the "Check In" button when browser is maximized.
- Fixed an issue that caused ClickOnce and protocol handler PuTTY to not load default session logging location.
- Fixed an issue that may crash the Web interface when a dependency forces check-in, prompting error "Global Catch: Secret requires checkout."
- Fixed an issue when adding parameters to SSH scripts where the dependencies are not updated to reflect the changes that are made. When populating the dependency details modal, a check is made to ensure the script's parameters match the parameters in its definition.

- Investigated an issue when RDP proxy is enabled causing a connection issue with using NTLMv1. NTLMv2 is required and is the default on Windows Server 2008 and later.
- Fixed an issue where database integrity monitoring sent false-positive alert emails for database rows changed by Secret Server.
- Fixed an issue where NULL NetBIOSName breaks IWA and causes spurious IP connections. The fix adds a step during the login process, specifically along the IWA path, that checks if the domain about to be checked has a NetBIOSName and, if not, fills it in.
- Fixed an issue in the UI with the drop-down menu with a predefined URL field values for secret template when creating a new secret.
- Fixed an issue in the UI where the incorrect folder menu was displayed on the favorites page.
- Fix issue where custom discovery account scanners that had "organizational unit" as their input where changed to "computer" if the scanner was updated.
- Fixed an issue when setting up a domain on a site that uses DE processing breaking Windows Authentication for that domain.
- Fixed an issue where the error message would not show for GCP and AWS scanner when the selected secret did not have a mapped password changer.
- Fixed an issue in the UI where "days until expiration" showed "expired" instead of a negative number to indicate time since expiration was met.
- Fixed an issue in the UI where the selected columns are only saved per folder, rather than across all folders. An "apply as default" check box was added to the column selector for folders. This allows the user to set up a default column setting for folder columns. It will be used when the columns are not set on a specific folder.
- Fixed an issue with live session viewing affecting Secret Server version 10.9.
- Fixed an issue where Connection Manager was not getting redirected to "get token" page when logged in with SAML.
- Fixed an issue that displayed an incorrect status message is displayed along with the error message. The "password verify was successful" message is no longer displayed along with the error message while checking heartbeat for invalid Active Directory password changes.
- Fixed an issue where the login screen was not remembering the last selected domain despite default login being set to "last selected."
- Fixed an issue with database integrity monitoring service Tablesnapshot exceptions not being handled.
- Fixed an issue where SSH proxy could fail and require a DE restart.
- Fixed an issue when saving an Active Directory domain, a 500 error on POST was prompted.
- Fixed an issue in the UI when using Internet Explorer that prevented adding a privilege account in a secret policy.
- Fixed a bug that was preventing searches on partial URLs from matching the secrets containing the partial URL. All parts of the URL host name, as well as any specified port, path, and query string are now parsed into terms according the indexing separators and are fully searchable in the UI and REST API.

- Fixed an issue where the "expire all secrets" and "delete all secrets" buttons were not present in reports in the new UI.
- Fixed an issue in the classic UI where a secret's URL path capitalization was being displayed incorrectly.
- Fixed an issue assigning a password on the secret if a username has consecutive special characters, such as `./` or `-_`.
- Fixed an issue that allowed local Secret Server users to log in without a synchronization group to Azure AD.
- Fixed an issue that prevented OUs from being added to the discovery scope with a message indicating the OU was already included due to its parent, which was invalid.
- Fixed an issue with updating associated dependencies when converting a secret template of a privileged account.
- Fixed an issue where a password change on RPC stays in pending status after meeting the retry threshold. We updated RPC functionality to stop the RPC, fix the RPC button to allow retry, and removed the pending password banner when the max attempts were reached.
- Fixed an issue with database integrity monitor not adding a separator for AggregateKey.
- Fixed a bug in extended search indexing that did not index all parts of the URL field, causing searches that contained non-indexed text to not return matches.
- Fixed a timing issue with Active Directory sync when processing results on a "synchronization now" event for groups.
- Fixed an issue where undeleting a secret did not honor the setting to not allow duplicate names.
- Fixed an issue where enabling proxy setting was not available in the secret policy with RDP proxy.
- Fixed an issue where OU exclusion does not allow different credentials underneath included a domain-specific OU.
- Fixed an issue with dependency bulk operations failing to run on privileged secrets.
- Fixed an issue with downloading system logs from the UI only listing what was shown in the UI. Implemented record ranges for all grids that implement the paging interface.
- Fixed an issue that caused an error when protocol handler MSI 6.0.0.33 did not overwrite the sslauncher registry key.
- Fixed an issue that prevented running a custom report in the UI.
- Fixed an issue where "view next password" was triggered when showing current password in the audit UI.
- Fixed an error when adding many groups or users to the folder permissions and saving.
- Fixed a permission to "view scanners link" in the menu UI.
- Fixed an issue when creating a secret in the UI it did not honor the default field values changed by the secret template.
- Resolved an issue that prevented RDP connections with non-standard port numbers when connecting via an SSH tunnel.
- Fixed an issue where scheduled task dependencies were disabled after an RPC on Windows Server 2008 R2 and 2012 R2.

Secret Server Cloud

Note: The following only apply to Secret Server Cloud (SSC). Other issues and features are assumed to affect both on-premises and cloud unless otherwise noted.

Azure ServiceBus Transport Type Setting

We added a new cloud-only setting called "Azure Service Bus Transport Type" in the global engine settings. This defaults to "Web Sockets," so the outbound port for engines communicating with SSC will only use port 443. Customers may change the setting to "AMQP," which changes the outbound ports to 5671 and 5672.

Other Issues

- Resolved an issue with Delinea One user synchronization.
- Resolved an issue that could cause repeated errors when accessing the OTP value for a secret.
- Changed some log entry types to warnings instead of errors.
- Fixed a bug that caused a SQL "incorrect parameter order" error when more than 2000 secrets were processed by RPC.

Known Issues

Important: If you encounter any of these known issues, please contact Delinea Support for assistance.

Note: These issues no longer apply to Secret Server Cloud as of December 19th.

- When used with Connection Manager, secrets that have a configured allow list for server connections will not display the allow list in the Connection Manager UI. This issue will be resolved in Connection Manager 1.4, scheduled for release on Dec 15, 2020.
- Secret Server on-premises installations may experience memory leak issues, which can be remedied by performing an IIS reset or application pool recycle.
- When creating a new user, the ability to select a domain is not displayed.
- The site connector for a local site cannot be changed in the UI. A workaround is possible via a manual edit in tbSite.
- Users configured for SAML are not correctly directed to the SAML login page when attempting to log in. Users will have to select "Local Login". This also applies to users that have MFA enabled in Secret Server—the user is incorrectly redirected after SAML login.
- SAML messages may not contain a necessary attribute, preventing users from successfully logging in via SAML. An available workaround is to select "Disable InResponseTo Check" in the Identity Provider settings of SS.
- Users with view permissions can no longer see generated OTP codes on secrets.

Secret Server: 10.9.000005/32 Release Notes

Release date:

- December 8, 2020 (on-premises version)

Critical Issue—Please Update to 10.9.000033 As Soon As Possible

We discovered a critical issue in Secret Server version 10.9.000032 shortly after releasing it. We encourage you to update to 10.9.000033 as soon as possible.

Important: If you have been affected by this issue, you may need to take corrective action in addition to updating. Please see below.

> **Note:** All the original notes for 10.9.000032, all of which still apply, appear in the "Secret Server: 10.9.000005/33 Release Notes" on page 1798.

Specifics appear below:

Issue

The issue occurs when managing local groups, users or roles in the User Management section of the New UI. This affects pages with URLs starting with `https://<Secret Server>/app/#/admin/user-management/…` When editing lists, the number of items in the list is erroneously limited to the number displayed in the UI. If the list previously contained additional items, they are removed when the list is edited.

Impact Examples

- When editing a local group, the number of users in the group is limited to 30.
- When editing the roles assigned to a group, the list of roles is limited to 30.
- When editing the roles assignments for a user, the list of roles is limited to 10.
- When editing the group memberships for a user, the list of groups is limited to 30.
- When filtering a group member list by domain and then editing the members, the member list only shows the filtered domain members, hiding the members from other domains.

Workarounds

If you cannot immediately update to 10.9.000033 or you have already been affected by the issue, please use the classic user interface when making changes to group membership, role assignment, or editing user roles or groups.

Secret Server 10.9.000032 automatically redirects attempts to edit users in the classic UI to the new UI, so you must directly navigate to the user management page by going to `https://<Secret Server>/users.aspx?legacyui=true` to edit them.

Versions Affected

Secret Server 10.9.000032 is affected.

Secret Server Cloud trials and new instances provisioned on or after 10 December 2020 are affected. Instances provisioned before 10 December 2020 are not affected.

Resolution

Customers impacted by this issue must manually restore their group and role assignments. Secret Server audit logs contain the required recovery information, such as which users were removed from a group. Use the classic UI (via the above mentioned URL) for doing this.

Notes

We removed Secret Server 10.9.000032 from our download sites.

Secret Server: 10.9.000002 Release Notes

September 22, 2020

Note: The system requirements last changed with version 10.7.000000. See Secret Server: 10.9.000002 Release Notes for details.

Upgrade Notes

Delinea encourages all customers to upgrade at the earliest opportunity.

Security

Security update to resolve a SQL injection vulnerability that an authenticated administrative user could exploit to achieve remote code execution on the Secret Server host system.

Common Vulnerability Scoring System (CVSS) v3.1 score: 8.0 (High).

CVSS v3.1 Vector AV:N/AC:H/PR:H/UI:N/S:C/C:H/I:H/A:H

Secret Server Cloud has been updated to include this security fix.

New Features and Enhancements

None

Bug Fixes

The following bug fixes apply to non-cloud Secret Server only. Secret Server Cloud has not been updated to include these fixes.

- Fix to Discovery rules to correctly handle OUs with bracketed names.
- Secret names in reports are now links to the corresponding secret.
- Logout from Secret Server no longer sends the `Clear-Site-Data` header, which could previously log users out of unrelated Web applications.
- SSH connections via SSH proxy now close correctly.
- Fixed an SSH proxy connection timeout when connecting via a distributed engine.

Secret Server: 10.9.000000 Release Notes

August 7-9, 2020

Note: The system requirements last changed with version 10.7.000000. See Secret Server: 10.9.000000 Release Notes for details.

Upgrade Notes

- Delinea encourages all customers to upgrade at the earliest opportunity.
- Secret Server now requires .NET 4.8 or later.

New Features and Enhancements

Alerts, Events, and Logging

- Added an event subscription for when a backup event fails or another exception is encountered.
- Changed log-level messaging in ClientToServerConnection to improve supportability.

API and Scripting

Added a parameter for an API endpoint to identify inactive users that may be members of a given group.

Authentication and Encryption

- Added validators to prevent passwords using keyboard sequences and patterns, dictionary words, and usernames.
- Added configuration setting to define the length of time to which users are locked out of Secret Server after the maximum login failure configuration setting is met.
- Created a more user-friendly experience when entering Time-Based One-Time Passwords (TOTP) generation keys.
- Added a password display option when viewing secrets using SSH Terminal.

Azure AD Synchronization

You can now synchronize users and groups in Secret Server with user and groups in an Azure AD. Does not require an on-premises AD for synchronization.

Connection Manager

Replaced the MacOS protocol handler. Secret Server no longer includes a MacOS version of the protocol handler for session launching. Instead, the Connection Manager free version is packaged with Secret Server for MacOS users.

DevOps Secrets Vault Integration

Added integration to allow Secret Server to create secrets in DSV and periodically push updates to those secrets. This allows customers to use DSV for fast API access and CI or CD pipeline integration while benefiting from the additional capabilities of Secret Server, such as credential rotation.

Discovery

- Updated discovery of scheduled tasks to no longer require Windows version compatibility between target and Secret Server's machine Windows version.
- Updated scriptable discovery to match all parameters to any field of a secret.

- Discovery can now be performed on AWS instances. This allows for OUs to be pulled from the AWS region scanner, machine detection from AWS Windows machine scanner, and machine detection from AWS SSH machine scanner.

Event Pipelines for Groups

Added triggers, filters, and tasks for group events to event policy pipelines.

Google Cloud Discovery

Note: GCP account and instance discovery requires that projects belong to an organization.

Implemented discovery across Google Cloud infrastructure including:

- Discovery and password changing of IAM service account users
- Discovery of instances associated to the projects
- Heartbeat and password changing of Google Cloud Platform (GCP) service accounts
- Token rotation for GCP service accounts

Heartbeat

- Added two values to the Web node health check to indicate when a node is in the process of an upgrade.
- Performance improvement: Added two advanced configuration settings for heartbeat consumer distribution across multiple nodes.
- Significantly increased heartbeat and RPC message publishing rates by allowing distributing work across nodes.

Integrations

ServiceNow integration now allows users to specify the ticket statuses that are accepted by Secret Server.

Launchers

Added a "Run Launcher using SSH key" configuration setting to secret policies. The selected secret will be applied to all PuTTY launchers attached to the secret.

LDAP Synchronization

Synchronize users and groups in Secret Server with users and groups in an LDAP directory.

Performance Improvements

- Added a new advanced configuration setting for the background worker process that requires higher publish rates. Setting configures the minutes to wait for the background workers to publish the secret to the queue before retrying.
- Improved efficiency of the expire secret background runner processes.
- Created a generic pre-processor for publishing batched secret lists to background workers.

- Implemented a performance enhancement to the FQDN lookup time per domain.
- UI now caches localization files for up to one hour, saving up to five seconds when the UI initializes.

RPC

Significantly increased heartbeat and RPC message publishing rates by allowing distributing work across nodes.

Security

Added a process to find lock keys that are over an hour old and subsequently remove them from the various caches.

Session Connector

You can now record video and keystroke data for sessions that do not use Delinea components at the user's client or target server.

Note: Removing the possibility of recording at a user's client or target server means that connections must be routed through a jump host running Microsoft RDS as part of the deployed PAM infrastructure. Connection recording occurs at a jump host running Microsoft RDS and additional Delinea software.

Session Recording

Changed behavior so viewing a session recording no longer opens a new tab in the browser from the UI.

User Interface

- Enhanced the new UI to present Admin > Folders via the left navigation panel and context menus. The dedicated Folders page is no longer available in the new UI.
- Added the ability for admins to set the default view used on the Admin page.
- Backup configuration tool and page now displays in the new UI.
- Launcher tool and page now displays in the new UI.
- Indexing Service page now displays in the new UI.
- Audit tab of the Admin Group Assignment page now displays in the new UI.
- About page now displays in the new UI.
- Discovery tool now displays in the new UI.
- New UI can no longer be disabled.
- Admin page can now preserve view selection.
- Discovery is no longer directly associated with the Active Directory page in the UI. Admin > Discovery is the updated location for discovery. "Directory Services" is the retitled page to manage Active Directory domains, as well as other directory services supported by Secret Server 10.9.

Bug Fixes

Alerts, Events, and Logging

- Fixed an issue where an event subscription is not triggered when a copy to clipboard event occurs.
- Fixed an issue where event subscriptions did not send a message when a login failure occurred due to two factor authentication failure.
- Fixed a verbose logging statement to correctly list "Windows Service" rather than "Scheduled Task" for service dependency scanning.
- Fixed an issue affecting the system event log entries for when Windows Authentication is enabled in IIS but is not enabled in Secret Server.
- Fixed an issue that displayed username (PII) in event logs when a password change was aborted.
- Fixed an issue that displayed incomplete information in the system logs when showing the Windows Account Login Failure error message.
- Fixed an issue where an Event Subscription or Event Pipeline did not trigger an event when copying a password to clipboard in the UI. Trigger is modified to be Password Displayed.

API and Scripting

- Fixed an issue with the SOAP API that could add permissions when InheritPermissionsEnabled is set. The call now removes inheritance when the permissions are updated.
- Fixed an issue that caused a 400 error when attempting to activate a license using API calls.
- Fixed an issue where the API was returning a server error rather than an access denied statement when making the call without the proper role assignment.
- Secret Server Cloud: Fixed an issue causing a failure when creating a secret in a folder via REST with a Secret Policy assigned.
- Fixed an issue with the API audit record statement for a password displayed event.
- Fixed an issue with the SSH exit command not working as expected.
- Updated tokens used for SSH fields in secret template extended mappings to match tokens used by the corresponding password changer fields: `$PRIVATEKEY`, `$PUBLICKEY`, and `$PASSPHRASE`. Properties on associated secrets can be consistently referenced using these tokens, regardless of whether the fields are mapped on the associated secret's password changer or on its template's extended mappings.

Authentication and Encryption

- Fixed inconsistent login issue when using SAML and two-factor authentication.
- Fixed an issue causing 'remember me' function to fail, prompting login after closing a browser window.

Checkout

- Fixed secret check in to no longer check password compliance immediately after check in. System now sets the compliance to pending, and the next time the compliance check happens it will be updated.

- Fixed an error in the workflow template for secret checkout when the reason field contained a large amount of characters.

Database

Fixed an issue with SQL not connecting due to high frequency schedules causing a timeout. Added a retry message to assist when a lock cannot be cleared.

Discovery

- Fixed an issue where the wrong field is passed to the dependency changer when a template field has a space in its name.
- Fixed an issue where the discovery host scanner was not substituting a token from an associated secret. Now, Discovery will try to match all parameters to any field of a secret.
- Secret Server Cloud: Fixed a sorting issue in the new UI on the discovery network view.
- Fixed an issue where accounts without an OU would be processed and removed by another discovery source.
- Fixed an issue where the Discovery page displays "please wait" if the site it is set to is disabled.
- Fixed an issue where some discovery dependency scanners were unable to be deleted within a domain.
- Fixed an issue that caused stack trace on discovery pages when making a copy of a template that has an inactive RCP mapping field.

Distributed Engine and Clustering

Fixed an issue where Distributed Engine would not reestablish a new connection after encountering an exception.

Export and Import

- Fixed an issue with CSV secret import using regex name patterns that caused import failures.
- Fixed an issue with importing secret to provide a more detailed error message to reflect the Secret Server environment used.
- Fixed an issue when exporting the user list where the exported CSV file did not contain a column for DisplayName data.
- Fixed an issue when exporting personal folders for users that had the same name. The fix introduces a toggle setting in the admin configuration of folders to allow the user to change their personal folders to either their display name or username and prefix.

General

- Fixed an issue with inactivity timeout.
- Fixed an issue in diagnostics where the internet connectivity check was still performed when disabled.
- Fixed a permissions issue for assigning secret policies to folders.
- Fixed an issue that caused an error when running asynchronous calls when reading and writing to CacheClient.
- Fixed an issue with secrets where proxying was enabled on secrets when RDP proxying was disabled.

- Fixed an issue where RDP proxy did not allow custom ports. Secret Server now checks the port number as appended to the computer name and uses the supplied port, if present.
- Fixed an access error for users with "view deleted secrets" role permission when converting a secret to a different template the original secret is not deleted.
- Fixed an issue where favorited secrets could not be removed from favorites after the user's access to the secret was downgraded to "list."
- Re-posted the clipboard Utility extension for Secret Server in the Chrome Web Store after policy changes made it temporarily unavailable.
- Added a setting to support web password filler's ability to autofill on multiple login page sequences and remediate users being redirected to the wrong URL. This launcher setting, by default, will send an extra parameter in the call to assist with multiple URLs.
- Fixed a memory leak appearing when the SSH port receives too many connection attempts in a short time. A third-party code library was not properly releasing memory used during failed connection attempts. Now, when an IP address makes too many (the default is over five) connection attempts over a specified time (the default is 30 minutes), that address is automatically denylisted in the database, and future connections are denied. When this occurs, notifications appear in the Secret Server logs. The solution also prevents targeted DOS attacks on the SSH port.

Heartbeat

- Fixed an issue where users with the edit permission were unable to perform a bulk operation to enable or disable heartbeat.
- Fixed an issue that displayed a misleading error on heartbeat failure. System now returns underlying error message when custom PowerShell password changer fails instead of a generic PowerShell error message.

Integrations

- Fixed an issue that caused issue with the TwoSense integration which causes Attempt User Password to fail.
- Fixed an issue with the scheduling task for the Quartz monitor.
- Fixed an issue where the site URL was not accepted on ConnectWise configuration if using http or https on the URL.

Installation and Upgrade

Updated the installer to configure worker roles to automatically start after a restart of Secret Server or IIS.

Launchers

- Fixed an issue where a secret containing a field named "Port" does not pass the value to the launcher.
- Fixed an issue that causes the browser to crash when a secret has a process launcher associated with session recording.
- Fixed an issue with the protocol handler not recording multiple processes when using the Web UI.
- Fixed an issue where protocol handler 6.0.0.23 launched sessions stopped shortly after launch when recording was enabled.

Remote Password Changing

- Fixed an issue with RPC not changing a dependency when running as a local admin.
- Fixed an issue with RPC where background tasks in the default task scheduler met a limit on active background tasks, causing a queue for new background tasks.
- Fixed an issue affecting new installations of Secret Server 10.7—the setting for "Change Password Using Privileged Account" did not appear in the new UI.
- Fixed an issue that could cause a duplicated password change when RPC is set to only change password when secret is expired.
- Fixed an issue when using a complex set of SSH commands for executing a password change by providing additional settings to accommodate various customer command needs.
- Fixed an RPC failure caused by replication latency for large Active Directory environments. Fix provides two additional advanced password types settings that allow for delaying and bypassing verification upon successful password change.
- Fixed an issue where Azure password changing failed during verification. Added two advanced password type settings for every password changer.

Reports

Fixed a display issue with pie charts not displaying correctly when the chart needed to display a large number of unique values.

Session Recording

Fixed an issue with session recording for SSH and RDP proxied secrets.

User Interface

- Fixed an issue in the new UI where the Secret List failed to reload and show values when adding a column containing a space.
- Fixed an issue where an error was produced when navigating from the Upgrade Secret Server page to the System Log page.

Webservices

Resolved IIS header conflicts for the X-Frame-Options header.

Secret Server: 10.8.000004 Release Notes

June 8, 2020

Note: The system requirements last changed with version 10.7.000000. See Secret Server: 10.8.000004 Release Notes for details.

Upgrade Notes

- **Delinea encourages all customers to upgrade at the earliest opportunity.**
- Security advisories are under review and will be published at the end of that review process. The link to that advisory will appear here.
- Delinea thanks Jay Huang from Insomnia Security for identifying the security issues leading to this release.

Security

High Priority Security Fix

Addressed incorrect user permissions validation.

- Common Vulnerability Scoring System (CVSS) v3.1 score: 8.8 (High).
- CVSS v3.1 Vector AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H.

Additional Security Fixes

- Added Custom "URL Security Check" to the Secret Server Security Hardening report. Delinea recommends configuring the Custom URL.
- Added host and port validation when using a proxied secret.
- Remediated potential cross-site scripting vulnerability.
- Modified user access controls to limit low-privilege application user access to administrative features.
- Implemented SHA-512 hashes for the launcher, replacing an older hash algorithm.
- Removed disclosure of internal IP addresses during authentication process of proxied connections.
- Modified when cookies are set during authentication.

Secret Server: 10.8.000000 Release Notes

April 3, 2020

Note: The system requirements last changed with version 10.7.000000. See Secret Server: 10.8.000000 Release Notes for details.

Upgrade Notes

- RMQ Helper was updated to install the latest RMQ version (3.8.2). See RabbitMQ and RabbitMQ Helper documentation for more details.
- Enabled the "port scan enable" configuration by default for discovery scanners.

New Features and Enhancements

RDP Proxy

Secret Server now includes an RDP proxy that facilitates third-party session management tools without exposing connection credentials. RDP proxy also enables optional keystroke recording for an RDP session without the need for the advanced session recording agent.

Event Pipelines

- Pipelines provide event-driven process automation within Secret Server. Users and administrators can save time by creating tailored workflows which are triggered automatically by events affecting secrets within Secret Server. Pipelines can be applied broadly across Secret Server or restricted by folder, secret template, group, site, or a combination of these and other criteria. For example, a pipeline could be configured to automatically change the password on an Active Directory secret when heartbeat fails.
- Infinite loops: By default, pipelines are configured to consider any event that executes five tasks within five minutes from the same trigger as an infinite loop. For example, "secret edit" is selected as a pipeline trigger, and "remote password change" is selected as the task. After the first edit is made on a secret, an RPC is triggered. Every time the RPC completes, a new edit is triggered, which, in turn, triggers another RPC. If this happens five times within five minutes, then an infinite loop is declared. If the RPC is slow, taking more than five minutes for five password changes to occur, then an infinite loop is **not** declared. In this case, use the "configuration advanced" page to change "event pipelines infinite loop time (minutes)" to a longer time.

Session Recording

Can now record multiple windows, including child processes, when session recording. Some applications used for session management open multiple GUI windows at once or launch child processes to render the session to the user. Secret Server can now record more than the main window of the parent process to ensure all user activity during a session is captured.

Web Password Filler

Note: This section covers the release notes for Web Password Filler (WPF) version 2.0.0, which is released with Secret Server 10.8.

This release includes updates for the following web browsers:

- Chrome
- Mozilla Firefox
- Edge (Chromium version)
- Opera

Enhancements:

- Web session recording is now supported in the WPF (please see the below section on Web Session Recording).
- Improved support for the refresh token. We improved the refresh token to better support SAML-configured Secret Server environments, and the WPF is updated to use this improved token. This also makes better use of the timeout settings for the SAML token. Secret Server
- Added timing restriction to the refresh button on the "sign in as" pop-up window in the WPF. This is to limit the number of calls that can be made in a 10-second time frame from going back to Secret Server to update the list of secrets.
- Added a new feature to match URLs by exact path. This option looks at the domain value in the URL and will only list secrets that have an exact match. During Web session recording, this option exactly matches the values

between `//` and the first `/` in the URL. For example, in `http://Company.Sub.Primary.Domain.TopDomain/subsite` it matches `Company.Sub.Primary.Domain.TopDomain`.

- Improved support for sites that have second-level subdomains in the URL, such as sites that have ".co.uk" or ".online.com.
- Added handling when fetching secrets from Secret Server for common second-level subdomains when filtering for applicable secrets on a website.

Bug fixes:

- Fixed an issue that returned a 500 error in the background when users tried to save a new secret (using WPF) when the new user interface is disabled in Secret Server. Secret Server.
- Fixed an issue that did not display the "Add Accounts to Secret Server" dialog if you entered credentials (that are not in a secret) into a site and try to automatically save it as a secret when "personal folders" are disabled for the Secret Server instance.
- Fixed an issue where not all folders were being returned when adding a secret, if the user had access to more than 1,000 folders.

Web Session Recording

- Added web session recording to the WPF browser extension. If a Web secret is configured for recording, WPF will now record the web session and display a recording icon at the top of the tab. Any additional Web browser tabs that are opened from that session (provided they stay on URLs that require recording) will also be recorded. All recordings will be available in Secret Server.
- If Web session recording is configured to run for a site, but the site prevents the recording icon from being placed in the browser tab's title bar, the WPF instead displays a pop-up message that the session is being recorded.

Connection Manager

Added two new settings for Connection Manager users on the Admin > Connection Manager page. Users with the "administer configuration" permission can modify these settings:

- Allow local connections
- Allow Connection Manager to save Secret Server user passwords

Note: These configuration options require Connection Manager 1.2 or later.

Integrations

- Added support for Devolutions Remote Desktop Manager integration, which allows users to retrieve passwords from Secret Server directly within Remote Desktop Manager. See https://updates.thycotic.net/secretserver/documents/SS-INTG-Devolutions.pdf. Secret Server
- Added support for WitFoo Precinct to integrate with SS, which allows users to view incident or event logs. See https://updates.thycotic.net/secretserver/documents/SS-INTG-Witfoo.pdf.

Smartcard Authentication

Added support for CAC/PIV smartcards pass-through authentication for RDP launchers.

Compliance and User Data

Personally Identifiable Information (PII) for inactive user data in Secret Server can now be anonymized to help organizations meet compliance regulations for GDPR.

Roles

- Added a new role permission, "session recording auditor," which allows a user to audit session recordings but not access the corresponding secrets.
- Added a new user role, "view secret password history," that allows users to have access to a password without granting them access to the password history. Users with edit or owner permissions on a secret who do not have this new role will be able to view and update the current secret password but will not be able to view previous passwords on the secret.

  Note: User roles that previously had the "view secret" permission will automatically gain "view secret password history" during the upgrade process.

Discovery

Added an extensible discovery "run as" privilege credential. Users can now specify a secret for the default credentials for running all PowerShell scripts on a site. This allows sites in different data centers to have different default credentials. This applies to remote password changing, checkout hooks, and discovery PowerShell scripts. If you want a specific secret checkout hook, secret password changer, or discovery scanner to use different credentials, you can still provide credentials in those areas, which take precedence over the secret set for the site. See "Discovery Overview" on page 522.

Performance Improvements

- Redesigned the "secret dependencies" page. Updates include database paging and filtering to fix page load performance issues in the new UI.
- The audit REST API was improved to increase performance when viewing the secret audit log in the new UI
- Added optimizations to improve secret search performance with very large secret sets. These include modifications to the secret list page, folder secret listings, search terms and other filters, the secret picker, and REST API endpoints. Endpoint improvements include options to not automatically count matching secrets and a method to count matching secrets rather than returning the set, both designed to remediate the overhead of secret filter set paging.

  Note: The optimizations may cause issues when upgrading, so they are not on by default. They may also cause issues with extremely long secret names (400+ characters). Please contact technical support for more information if you believe you might benefit from the optimizations.
- Altered the MS SQL Server collation used for secret name search to improve performance. As a result, secret names starting with numeric values now return before secret names starting with symbols. An advanced setting (contact Delinea Support) is available to revert this back to the original sort order at the cost of some of the performance gains.

Security

- Fixed an issue where a local user could log in to the Secret Server SSH terminal despite an expired password.
- Fixed a reflected cross-site scripting (XSS) vulnerability.
- Extended the default HSTS max age to one year and extended it to cover subdomains.
- Fixed a cross-site scripting (XSS) issue.
- Fixed an issue preventing a security header from being returned in certain locations within the application.
- Secret Server no longer allows any cross origin (CORS) calls.

API

- The rest API documentation has a new look and feel, including a table of contents.
- Added 67 new endpoints. See The "REST API Reference Download" on page 1500 for details.
- Added the ability to autogenerate API clients for PS, C#, Java, Postman, and Insomnia.
- Added PATCH API call methods, which are designed to update only values that are passed through the call. New object types include:

  ```
  /secrets/{id}/email/secrets/{id}/general/secrets/{id}/security-checkout/secrets/
  {id}/security-general
  ```

  An example of a PATCH call is:

  ```
  PATCH/secrets/{id}/general{{secretPolicy: 1}}
  ```

  or more descriptive:

  ```
  PATCH/secrets/{id}/general{{secretPolicy: {value: 1, dirty: true}}}
  ```

Logging

- Added verbose logging for:
  - Active Directory password changers
  - SQL Server password changers
  - MySQL password changers
  - Oracle password changers
  - Sybase password changers
  - LDAP Privileged password changers
  - Mainframe password changers
  - Windows password changers
  - SSH password changers
  - AppPool dependency
  - SSH dependency

  - SQL dependency
  - PowerShell dependency
  - ComPlus dependency
  - FlatFile password changers
  - ScheduledTask password changers
  - WindowsService password changers
- Added a correlation ID to every request generated by new UI pages. This ID is tracked by the API and is included in the server log when relevant. To get the IDs, users can go to the new UI diagnostics page to download a log of all HTTP requests made within the past two days.
- Updated error handling to process unknown errors as "UnknownError." Previously unknown errors logged as "LoginFailed."

Cloud

- Fixed a caching issue where discovery scanning sometimes did not find an existing organizational unit (OU) for SSC instances.
- Significantly improved shutdown and restart times for engines connected to SSC.
- Added a new configuration setting to allow SSC admins to opt non-admin users out of seeing the SSC upgrade notification banner before a release. Access this "display maintenance message To administrators only" setting at Admin > Configuration > General tab > User Experience.
- Removed the legacy Web Password Filler bookmarklet.

General

- Updated the layout for configuration pages to include more details about page sections.
- Removed options such as "import secrets" and "launcher tools" from the grid icon menu. All menu items are available from the admin page.
- Redesigned the "user preferences" page. Click the profile icon in the upper right, and then select User Preferences.
- Added actions that allow a user to cancel an existing request in the request log grid.
- Added auditing for changes made to discovery source pages.
- Application requests made from the API can now be approved or denied on the inbox page.
- Added a checkbox to the settings of the scheduled task and Windows service scanners to use NETBIOS name instead of UPN when matching domains.
- On secrets that require both check out and comment, users are no longer required to submit two comments. We combined the check out and comment prompts.
- Added a "next password reveal" audit for secrets that are manually updated with a next password.
- Converted active templates to new UI.

Bug Fixes

- Fixed an issue to Web Password Filler to have the URL encoding match what is configured in the secret. Issue related to handling of ampersands contained in the launcher URL.
- Fixed an issue where request access emails did not send to some approvers.
- Fix to address accuracy of login time stamps when using SAML Active Directory Federated Services.
- Fixed an issue where copying custom reports sometimes did not allow a user to save.
- Fixed an issue where some secrets did not display when picking associated secrets for a secret policy or event subscription. Associated secrets on secret policy edit will now default to not filtering by password type configured on the template. The "show all" button will also no longer filter by this setting.
- Fixed an issue where folder permissions could be changed via REST API despite the "inherit permissions from parent" setting being enabled. Changes to folder inheritance through the API will now be reflected in the UI and will be messaged in the API.
- Added a new advanced configuration advanced setting, OffloadRunNowToDifferentNode. If true, offloads the "run now" commands to nodes with background worker enabled. This only applies if Secret Server nodes have different RabbitMQ site connectors.
- Fixed an issue where filter parameters were ignored when using `filter.searchField` in the API method. Users are now able to perform partial searches through the API.
- Fixed an issue where users with edit permission on a folder were unable to perform the "convert template" bulk operation on secrets they owned within that folder.
- Fixed an issue where deleted engines were still shown on the engine status widget.
- Fixed an issue where RADIUS two-factor authentication access challenge prompts did not work with Integrated Windows Authentication.
- Fixed an issue where the time format in the secret access request form page did not match the default time format in Admin > Configuration.
- Fixed an issue in the new UI where the dependencies tab was not displayed correctly.
- Added a notification message when a secret is forced to check in by another user.
- Fix made for RPC password changes getting stuck in the processing state, after disabling RPC.
- Fixed an issue where a folder name is truncated to 50 characters when imported via XML.
- Fixed an issue where the audit log showed duplicate view instances for secrets
- Fixed an issue where Unicode characters would display as question marks on the SSH proxy session data page.
- Enhanced search logic so that an exact match for folder or secret name will appear first in the search results list.
- Fixed an issue where proxied sessions (RDP or SSH) did not properly terminate if session recording was previously disabled for the secret.
- Fixed an issue where the "inherit permission from folder" setting was disabled when sharing a newly created secret in the old UI.

- Fixed an issue where other users' personal sub-folders were still visible to users who did not have View permissions on them despite the "require view permission on specific folder for visibility" setting being enabled. This only occurred on the Admin > Folder Permissions & Audit page.
- Fixed an issue where different time zones were shown when editing the RPC settings of a secret.
- Fix made to address error when attempting to export diagnostic logs which resulted in incomplete records.
- Fixed the "value cannot be null" error message after clearing a report filter.
- Fixed an issue where the `POST /secret-templates/generate-password/{secretfieldId}` endpoint did not properly handle password fields.
- Fixed an issue where the web password filler could not login local users through SAML authentication.
- On the security hardening report page, increased the maximum size of the zero-information disclosure message.
- Fixed an issue where AD sync processed a disabled domain.
- Updated the distributed engine "processing" status query to exclude inactive secrets in its data count on the SiteView.aspx page.
- Fixed an issue where duplicate secrets could be linked to a discovery scanner.
- Fixed an issue where edits to a dependency modal did not save.
- Fixed an issue where the "automatic user management" setting did not re-enable Integrated Windows Authentication (IWA) users properly.
- Fixed an issue with attaching files while creating a new secret that had check-out or require comment enabled.
- Fixed an issue with the REST API `POST /secret-templates/` endpoint, which did not work.
- Fixed an issue in folders that contain over 30 sub-folders where clicking to maximize the folder list threw an error.
- Fixed an issue where converting a secret template failed after modifying the secret template without first saving the template.
- Fixed an issue in the new UI where require approval logic on a secret policy blocked users from creating secrets in a folder where the policy was assigned.
- Fixed a new UI issue where copying a secret template failed when using the Internet Explorer browser.
- Fixed an application error message when secret access was revoked by its owner, while another user was attempting to check it out.
- Fixed inaccuracies displayed in "last accessed" and "created" column data in secret grids.
- Fixed an issue in the way IP addresses in URL fields were indexed and how indexed searching was conducted. Searching by partial IP address now correctly matches against URL fields for any substring of the IP address, as long as the search term starts with a whole octet of the field.
- Fixed a display issue for drop-down grid menus in the new UI.
- Fixed an issue when sharing a secret where a message incorrectly stated that the user would no longer be owner of the secret.

- Fixed an issue where duplicate groups populated when selecting from the "add Group/user" dropdown on the Admin > Groups page.
- Secret folder tree expansion is no longer lost when editing a folder.

Secret Server: 10.7.000059 Release Notes

December 9, 2019

Note: The system requirements last changed with version 10.7.000000. See Secret Server: 10.7.000059 Release Notes for details.

Upgrade Notes

- Fixed an issue with Legacy SAML. Customers are encouraged to migrate to Secret Server's updated SAML if still using the Legacy version.
- This release launches a new web password filler. To update your web password filler extension, go to the extension download site for your browser and platform.
- Customers with multi-node environments using Advanced Session Recording will need to update all ASR agents after your Secret Server upgrade to take advantage of the RabbitMQ failover updates in this release. We recommend doing so, but not doing so will not impact current functionality. See the "RMQ Failover" note below and the Secret Server Advanced Session-Recording Agent Installation KBA.
- Users will be directed to the dashboard Overview tab for their first login after upgrading.

New Features

Data Retention

Secret Server now allows administrators to permanently delete audit records for tables that either contain Personal Identifiable Information (PII) or tables that can grow large in enterprise environments. To configure these settings admins need to add the permission "Administer Data Retention" to the user's role and then the user can navigate to **Admin > Data Retention**. Only users with the "Unlimited Administrator" permission can assign this new permission.

Manual Rolling Upgrades

A new "Manual Rolling Upgrade" feature is available when upgrading from Secret Server version 10.7.000059 or above. Using this process, customers using clustered web nodes with a load balancer can experience little-to-no downtime during the upgrade process, but this process requires manual steps by an admin with Web node and database access.

RMQ Failover

Updated Secret Server to support durable exchanges for RabbitMQ (RMQ). This allows clustered site connectors to failover without impacting Secret Server's processing. Distributed engines will auto-update after Secret Server's upgrade to also support durable exchanges through RMQ.

Note: Older Advanced Session Recording Agents (ASRA) can be used with this version of Secret Server but ASRAs will not benefit from this change to failover handling. To include failover capability for ASRA an updated agent must be deployed.

Technical Details: The ExchangeDeclare logic in MessageQueue client was altered to attempt to create durable exchanges with logging. A durable exchange is automatically re-created if RabbitMQ restarts for any reason. Non-durable exchanges disappear when RMQ goes down and can only be re-created by some external action. If the new logic detects that creating the durable queue failed, it will log an error and attempt to create a non-durable queue.

Note: The presence of legacy non-durable exchanges can prevent the automatic creation of durable exchanges. See "RabbitMQ Durable Exchanges" on page 787

Time-Based One-Time Passwords (TOTP)

Added a new feature where Secret Server can now generate time-based one-time passwords (TOTP) for web secrets. This allows users to implement TOTP on shared secrets. Configuring secrets for TOTP begins at the secret template level.

Truncated Log Data

Added the ability to truncate table logs for several types of data that log to the "Status Message" table. These messages can contribute to excessive log data and slow performance. The option to truncate each message type is called "Days to Keep Operational Logs" and is under the "Advanced" sections on the following list of configuration pages. Minimum message retention time is one day and the default is 30 days. The logs include:

- AdminDiscovery.aspx (**Admin > Discovery**)
- AdminSearchIndexer.aspx (**Admin > Search Indexer**)
- ConfigurationActiveDirectory.aspx (**Admin > Active Directory**)
- ConfigurationPasswordChanging.aspx (**Admin > Remote Password Changing**)
- ConfigurationSshProxy.aspx (**Admin > Proxying**)
- ConnectWiseConfiguration.aspx (**Admin > Folder Sync**) Setting only available when using the "Database" Folder Synchronization Method on this page.

Technical details: A background task was added that scans the status message table every 12 hours and checks the status messages against configured values for how long they should be retained. These configured values were added to applicable UI pages.

Web Browser Plugins

The Web browser plugins for Secret Server were rebuilt with a new look and feel and now have additional browser and site support. These new plugins are available for:

- Google Chrome
- Mozilla Firefox
- Microsoft EdgeKeep Secret Name History
- Opera

These features from the old browser plugins have been improved to allow more flexibility:

- Create secrets
- Select secret template
- Generate complex password

Users can now authenticate to Secret Server directly from the Web plugin, including support for 2FA options, such as DUO. Log in via Secret Server is also available for users with single sign-on, SAML, or other multi-factor authentication mechanisms. Web plugins automatically identify manual entry of new credentials in a Web page and offer to save the credentials as a secret. There is also improved support for sites that use multi-page login mechanisms.

Enhancements

Advanced Session Recording

Added a new setting to disable keystroke data from advanced session recording metadata. The new setting is called "Default Keystroke Recording Configuration" and can be configured under **Admin > Configuration > Session Recording > Configure Advanced Session Recording.** Click **Collection name** to edit individual collection settings or agent settings. By default, advanced session recording keystrokes are enabled.

Remote Password Changing upon Regex-Defined Error

Added a new regex setting to automatically retry a remote password change (RPC) with a regenerated password if the original RPC failed due to a specific type of error.

Go to **Admin > Remote Password Changing**, click **Advanced** under the **Configure Password Changers** section. The new setting is **Attempt Password Change with new password when error contains (regex)**. Edit it to provide the regex failure code that will trigger the automatic next password RPC.

Discovery

Added messaging for when computer or dependency scans do not run due to having no scanners configured for a discovery source.

Distributed Engine Offline Status

Updated the definition of distributed engines' offline status to be the configured heartbeat interval times three. For instance, if your heartbeat interval is configured at 5 minutes, the engine will report offline if Secret Server and the engine do not successfully communicate within a 15-minute time period. Engine online and offline states were also added to subscription actions to allow notification to admins when engine states change.

New User Interface

- Redesigned the Admin landing space. Click **Admin > "See All"** to explore the new layout.
- Redesigned DoubleLock.
- Added new "Recent Activity" section to the Home dashboard page to display recent activity at a glance.
- Updated the Security Hardening tab in the Reports page.
- Updated the IP Address Management pages under Admin.

- Added custom logos. Added custom "full-sized" and "collapsed" logos for the new UI in **Admin > Configuration** under in the **User Interface** section.
- Added dark mode theme option in the new UI. To change theme mode preferences, go to **Account Settings > Color Mode**. Options include Light Mode, Dark Mode, or Default (mode will update based on user's OS color mode settings).
- Added a new setting to configure the inactivity time before the new UI goes into dark screen "sleep mode." To configure go to **Admin > Configuration > User Experience > UI Inactivity Timeout**.
- Converted the Groups page to the new UI.
- Updated error messaging in the new UI to display folder synchronization and deletion errors.
- Updated the date picker to allow for future start dates and time selection without first adjusting the end date when requesting secret access. End dates are automatically adjusted to align with the start date +1 hour.
- Updated grid downloads in the new UI to download according to new options. User options now include choices to download all data or specific rows of data, and specify date format. You can also choose time zone options of UTC, server time zones, or the local browser time zones.

  Note: for downloaded reports users' time zone options are limited to UTC or the server time zone.
- Updated behavior of new UI so that clicking the "Select All" check box at the top of a secret grid selects all rows. Previously the check box selected only the items currently loaded on the page.
- Added the "View Audit" button to the reports page of new UI.
- Added the "Upgrade Available" banner to display in the new UI.
- Added the ability to drag-and-drop child folders into the root folder. Folders will automatically re-order alphabetically in the left navigation pane.

  Note: This action is only allowed if users have the "Create root folder" permission and own folders that they are attempting to move.
- Added folders to the "Shared With Me" page.
- Added new inbox notifications including "getting started" notifications for new installs and administrator alerts when an instance is close to hitting licensing limits.
- Added the ability to mark Inbox notifications as read or unread for most notification types.
- Added the ability to browse by folder name using the URL format `[SecretServerURL]/SecretServer/app/#/lookup?folderPath=[FolderName]`. If multiple folders exist with the same name, this URL search schema only directs users to the first folder listed within the left navigation pane.
- Updated Favorite star icons to remain in column view when the Name column is resized.
- Expanded file-size allowance on file uploads. File uploads can now be up to 10 MB.
- Grid results updated to auto-load 30 results instead of 15.

Licensing

A second distributed engine is now available, by default, for the local site.

Reports

- Updated several reports to no longer show deleted secrets.
- A new out-of-the-box report called "Secret Templates without an expiration field" was added to display any secret templates that have a password field but do not have an expiration field set.

Secret Template Import and Export

Updated secret template settings for importation and exportation to include:

- Is Required?
- Edit Requires
- Hide on View
- Secret template icon
- Keep Secret Name History
- Validate Password Requirements on Create/Edit
- Field Slug Name
- Type Description
- One Time Password settings

The secret template settings that do **not** transfer include:

- Launcher settings
- Password changing settings
- Session recording enabled
- Associated secrets

SSH Proxy

- Updated "connect as" to accept key-based SSH authentication without also requiring a manual password.
- For SSH proxy sessions, added the option set:
  - Only record keystrokes
  - Only record video for sessions.

  By default new installs will only record keystrokes on SSH proxy sessions to preserve disk space. To configure this setting go to **Admin > Configuration > Session Recording tab > Secret Server Proxy Session Recording**. Edit the **SSH Proxy Session Recording Options** dropdown list. The options include:
  - Record keystrokes and video
  - Record keystrokes only
  - Record video only
  - Do not record

Verbose Logging

Added Verbose Logging for:

- AWS password changers
- AWS discovery scanner
- ComPlus dependency scanner
- PowerShell discovery scanner
- Flat file discovery scanner
- ODBC discovery scanner
- SSH discovery scanner
- ESX discovery scanner

Terminal

- Added terminal instructions for how to view SSH proxy credentials in the new UI under **Secret Options > Show SSH Terminal Details**.
- Removed restrictions from the allowed number of concurrent logins for SSH terminal. Previously, terminal logins were tied to the "Maximum concurrent logins per user" setting that establishes this number for UI-based users.
- Added Unicode support for SSH command menu items (names and descriptions).
- Added "clear" command to terminal.

Database SQL Indexes

Added new SQL indexes for the following areas:

- Column LauncherSessionGuid on the Launcher Session Video (tbLauncherSessionVideoSegment)
- Event Queue Monitor (tbEventQueue)
- Expired Secret Monitor (tbSecretDependency table)
- Folders (tbFolder, tbFolderGroupPermissions)
- General Navigation (tbUserSession)
- Launcher Activities (tbSecretSession)
- Log In (tbUser)
- Node Activation Check (tbNodeLicenseActivation)
- Secret Log table (tbSecretLog)
- System Reports (tbAuditUser, tbAuditSecret)

Unique Field Slug IDs

Added a new "Unique Field Slug" ID column for secret templates to allow users to create secrets with duplicate field names without compromising the ability to target each field name with a unique identifier for API calls.

User Variables for Scripting

Added the following user-based script variables to be used in API calls as arguments:

- `$SECRETSERVERUSERID`
- `$SECRETSERVERUSERNAME`
- `$SECRETSERVERDISPLAYNAME`
- `$SECRETSERVEREMAILADDRESS`

This allows, for example, that when a specific user runs a check-out hook, they can pass a user email, ID, username, or display name as a parameter into the script to use a check-out hooks and related AD functionality in Secret Server through the API.

API and Scripting

API General

Added a setting that allows users with view permission on a secret to get the secret's "autoChangeNextPassword" field in the API. This setting is enabled under **Admin > Configuration > Permission Options**. Set **Allow View User To Retrieve Auto-Change Next Password** to **Yes**.

New API Calls

- Get one time password code and seconds: `GET /one-time-password-code/{id}`
- Search secrets by URL: `POST /secret-extensions/search-by-url`
- Get AutoFill values for URL by secret ID: `POST /secret-extensions/autofill-values`
- Update secret field: `PUT /secrets/{id}/fields/{slug}`
- Update secret: `PUT /secrets/{id}/restricted`
- Get SSH Terminal details: `POST /secrets/sshterminal`
- Get extended regex values by secret: `GET /extended-fields/regex/{secretId}`

Removed API Calls

- Search app clients: `GET /app-clients`
- Update secret: `PUT /secrets/{id}`
- Update secret field: `PUT /secrets/{id}/restricted/fields/{slug}`

Integrations

- Added support for Open ID Connect to integrate with Secret Server authentication.
- Added Connection Manager as a tools option in the grid menu and under the Admin space in the new UI.

Performance Improvements

- Added server-side paging to reports in the new UI to address performance issues when attempting to load reports with large numbers of records.
- The new UI will no longer load the subfolders If a parent folder has more than 30 subfolders within it on the grid page. Instead, a folder picker will display above the folder's secrets that will allow users to select a specific subfolder.
- Applied enhanced SQL querying logic on the groups pages so that environments with large groups no longer experience page timeouts when processing group data.
- Improved the shutdown performance in distributed engine.
- Removed the welcome widget from the dashboard on the classic UI due to page load issues in large environments.
- Enhanced SQL query for the unlimited admin report to improve performance for large environments.
- Added a new "use database paging" setting for the custom reports page. Database paging allows the database to load large reports more quickly. We recommend database paging if the query is expected to pull large amounts of data for the report. Implementing database paging may not work if the SQL query uses some keywords, including TOP, OPTION, INSERT, UNION, WITH, or aliases containing the word FROM.

  Example queries:

  - Works using database paging: `SELECT * FROM tbSecret WHERE NAME LIKE 'Test%'`
  - Does not work using database paging: `SELECT TOP 10 * FROM tbSecret WHERE SecretName LIKE 'Test%'`

Security

- Updated PuTTY to version 0.73. Updated version addresses several PuTTY vulnerabilities, including one critical and two high severity items. CVE-2019-17067, CVE-2019-17068, CVE-2019-17069
- Addressed a vulnerability with the SDK client account handler.
- Fixed a permissions issue in the new UI where password requirements did not obey the "administer custom password requirements" permission.
- Added audits and event subscriptions for viewing passphrases and SSH keys.
- Addressed a Remote Code Execution (RCE) vulnerability that allowed parameter changes for an action without validating user permissions.
- Resolved an issue for SSH scripts and SSH remote password changers where sensitive information was being written to log files:

  - SSH remote password changers will now only log the comment for each command as it runs.
  - SSH scripts will only log that they ran because they have no comment for each command.

  Note: If you manually test an SSH script or password changer, the full output will still be shown for debugging purposes, because you just entered the credentials yourself.
- Resolved a URL redirection vulnerability.

- Added configurable parameter quoting for custom launchers.
- Resolved three cross-site scripting (XSS) vulnerabilities.
- Fixed an XML external entity (XXE) injection vulnerability.
- Removed user information that was returned in an API call.
- Added auditing for changes made to the session recording configuration page on the **Admin > Configuration > Session Recording** tab.
- Added auditing for test script actions in the **Custom Command Edits** section in the **Admin > Scripts** pages.
- Added auditing to the **Admin > Configuration > Ticket System** tab. Audits are logged under **Admin > Configuration > General tab > View Audit.**
- Updated missing secure cookie attributes when "Force HTTPS" is enabled.
- New installs running 10.7.000059 or later will now automatically apply zero information disclosure.
- Added SHA1 and SHA256 hashes for protocol handler.
- All Delinea DLLs and EXEs are now signed with the Delinea Software certificate including distributed engine, advanced session recording agent, and MemoryMQ applications.

Bug Fixes

- Fixed an issue where creating folders through the API failed to set a secret policy.
- Fixed a memory leak issue where leaving Mac launcher sessions open for extended periods of time consumed increasing amounts of memory on the machine hosting the session. This issue was incorrectly believed to be fixed in the Secret Server version 10.7.000002 release.
- Fixed an issue where an access approval email link did not work if Integrated Windows Authentication (IWA) and two-factor authentication were enabled.
- Fixed an issue where Unix secrets were not reported in the "Password Last Changed" report because the Unix Account template did not have a password expiration field by default. Unix password fields are now set to expire at 30 days by default.
- Fixed an issue where pressing Enter with the cursor in the Search bar on the Discovery Network View page would open the create rule dialog.
- Fixed an issue where new users were not adequately loading in the dropdown option for subscribers in the "discovery rule alerts" setting if users increased from under 40 to over 40 users.
- Added clear error and validation in entering credentials for a discovery scanner.
- Fixed an issue where localized language customization did not apply to some product pages due to default cache keys or inconsistent HtmlHelper methods implemented on those pages.
- Fixed a bug where some pages in the old UI did not follow customized headers from CSS stylesheets.
- Fixed an issue where extended search terms were not applied for URLs. Updated BookmarkletSecretSearcher so that it will not do extended hashes on URLs that might result in many erroneous matches. For instance, facebook.com would match face.org.

- Fixed issue where non-AD discovery sources (ESX, PowerShell) would not match dependencies with domain to an existing AD Domain. If the dependency scan item has a field called "domain," it will attempt to map to an existing domain.
- Fixed a bug where the REST API "daysUntilExpiration" field returned a blank value when calling for a secret summary.
- Added a query to resolve sort ordering issues for dependency numbering.
- Fixed an issue in the new UI where deleted secrets remained on display on the favorites widget in the home tabs.
- Fixed a "bad token" error on login for mobile apps (Windows Desktop, iOS, or Android) for local users.
- Fixed an issue where SSH keystroke data was not searchable from the session playback page due to proxy session data not being correctly hashed in the database.
- Fixed a ticketing system bug where the option to require users to either provide a ticket number or a comment for requesting access incorrectly required both a comment and a ticket number to access the secret. This issue existed when requesting access to the secret through the UI, workflows, or terminal.
- Fixed a bug where hashed terms were intermittently slow to return search results in the new UI.
- Fixed a rendering issue for the group edit page when using the IE browser.
- Fixed a localization issue in the SSH command menus where setting non-English as the global or user preference threw an exception when trying to save a secret policy.
- Fixed an issue where an "object disposed" exception was thrown when navigating away from the new UI soon after application pools were recycled. This occurred because of an incorrect re-use of a service that is for processing the first Web request to the application.
- Fixed a bug in the new UI where personalized preferences for launcher settings on a secret were not allowing users with view access to save.
- Fixed a bug where custom columns on grids in the new UI occasionally tried to display the same column twice and threw an error.
- Fixed an issue where a syslog header was missing the hostname when logging from an engine.
- Fixed an issue where Duo authentication was not checking for valid fallback device options when the configured "default device" failed to authenticate.
- Updated the "find new dependencies" feature to be available to users with edit permissions on the secret. Previously the new UI required users to have owner permissions.
- Resolved a timing issue where secrets with scheduled password changes enabled would get erroneous heartbeat fails due to remote password change (RPC) expiration and heartbeat occurring at the same time. Now heartbeats are skipped for an interval of five minutes before and after a scheduled password change to allow password change completion before heartbeat is attempted.

Secret Server Release Notes 10.7.000002

October 8, 2019

Note: The system requirements last changed with version 10.7.000000. See Secret Server Release Notes 10.7.000002 for details.

Note: Secret Server 10.7 and later no longer support using SQL Server 2008 R2 as the database for Secret Server.

General Bug Fixes

- Fixed an issue where automatic password changes (including scheduled password rotations) did not run on environments with distributed engines.

  Note: Using the "Run Now" button (Admin > Remote Password Changing) to manually start the process was unaffected.

- Fixed an intermittent PuTTY crash with the RDP Launcher caused by PuTTY version updates in Secret Server not properly disposing named-pipe-related entities after the 10.7 release.
- Fixed an issue where writing to the local CEF log file threw exception errors, causing deadlocks, due to a missing allowlist for local CEF logging.
- Fixed an issue where the "Generate New SSH Key" button did not generate a public key when creating a new SSH key secret.
- Fixed an issue where disabling two-factor authentication using the "Lost Phone" option did not trigger a "Two-Factor Changed" event, resulting in the event logs not recording the reset of the user's two-factor authentication mechanism. Added two new events, one for a successful and one for an unsuccessful TOTP reset.

Mac Launcher Fixes

- Fixed an issue where copy actions in a launched session would drop the last letter in the copied string.
- Resolved an issue where a Mac RDP session could crash if left open for a long, inactive period.
- Fixed an issue where right-clicking within a PowerShell window terminated the Mac RDP session with the server.
- Fixed an issue where turning off custom resolution in "Launcher Settings" caused a crash on next launch.
- Added a check to ensure a non-zero height and width are used. If the width is set to 0, then a width of 1024 is used. If height is set to 0, then as height of 768 is used. These were the previous default settings.
- Enhanced messaging when sessions are disconnected due to server-side tasks. The user now receives a dialog notification letting them know that the session was closed from the server.
- Fixed an issue where pressing the Option key during a Mac RDP Launcher session malfunctioned as if the user was indefinitely holding down the key.
- Added scrollbars to the FreeRDP window to improve mouse scrolling. The mouse wheel now scrolls windows inside the connected server but does not scroll the FreeRDP window. Users must drag the FreeRDP scrollbars to move the content.
- Made window resizable down to 100×100 pixels and up to the specified dimensions.
- Fixed a memory leak where leaving Mac launcher sessions open for an extended period consumed ever-increasing memory on the machine hosting the session.
- Fixed a session recording issue in the Mac Launcher when multiple recorded sessions occurred at the same time. If a recorded session began while another session was already recording, the original recording would not be saved.

Secret Server Release Notes 10.6.000027

Secret Server 10.6.000027 Release Notes

This is a security release for Secret Server On-Premises. Released on June 18, 2019.

Security Advisory

Bug Fixes

MS SQL Server 2008R2

*	Fixed an issue for MS SQL 2008R2 users that caused search to fail in the New UI.
*	Fixed an issue for MS SQL 2008R2 users when loading the "Manage Secret Access Requests" page from updates in the New UI.

Secret Server Release Notes 10.6.000026

The second minor release after 10.6 for Secret Server On-Premises. Released on May 29, 2019.

Upgrade Notes

•	After upgrading to 10.6.000026, the first admin who logs in to Secret Server is prompted with a wizard that walks through the new user interface (UI) configuration settings for their instance. Users can adjust these settings at any time from **Admin > Configuration** in the **User Interface** section.
•	Enterprise customers who use load balancing with RabbitMQ may want to delete queues after upgrading Secret Server from versions earlier than 10.6. If customers using RabbitMQ do not delete their queues, they will **not** lose functionality, but old queues that were renamed in a 10.6 architectural update will continue to fill up with messages.

New Features

New User Interface

•	In the 10.6.26, Secret Server's user interface was updated, featuring a black left-hand navigation pane. The look and feel and user experience updates were based on user feedback from the beta version.
•	As noted in the upgrade notes section, after upgrading the first admin who logs in to Secret Server is prompted with a wizard that walks through the new UI configuration settings for their instance. Users can adjust these settings at any time from **Admin > Configuration** in the **User Interface** section. After settings are saved, users with access to the new UI are shown a short demonstration of switching between new and classic user interfaces upon their first login.

We are incrementally introducing feature parity between the new and classic UIs. In this upgrade the new UI added support for:

•	Emailing reports to users.
•	Deleting reports.
•	Converting secret templates.
•	Uploading files to secrets.
•	Copying and converting file attachments.
•	Setting automatic password change rotations when creating new secrets.
•	Rotating SSH Keys and Passphrases for SSH secrets.
•	Rotating SSH Keys and SSH passwords using a bulk operation.
•	Displaying custom columns in the folder view.
•	Updated the Secret Picker to include a folder tree browser when selecting a privileged secret for Remote Password Changing.
•	Added a password complexity indicator that displays when editing a Secret's password.
•	Updated Search functionality to include folders.
•	Users can now "favorite" folders by clicking stars on folders in a table view or by right-clicking folders in the left navigation pane.
•	Additional "View" audits were added to the new UI to track users who view Next Passwords, SSH Keys, and Passphrases.
•	Optimized SQL queries when searching for secrets in the new UI. This includes: Faster searches, the paging and sorting is now performed in the database instead of the web server Ability to filter by secret permission, extended type id, password type id, RPC enabled, recently used date range Can now return extended fields Sorting by an invalid field now returns an exception instead of sorting by the secret ID

Advanced Session Recording Without Launcher

•	Added video recording capability to the Advanced Session Recording agent, with a new configuration setting that can enable recording of a video on an endpoint even when a launcher is not used to begin a session.
•	Advanced "headless" session recordings will show a 'Session Minimized' notification for periods during which the session was minimized, or otherwise hidden from the user such as during the final stages of logging out.

•	To prevent unnecessarily long session recording sessions, added a "Max Session Length" setting for recorded videos. The default for this setting for on-premises instances will stop a session after 24 hours of no activity. Administrators can set this to a maximum of 48 hours.

New Secret Templates

•	Added a new "No Password" template for SSH/Linux Key rotation. These templates allow users to rotate SSH and Linux keys without the password field requirement that exists on other Unix secret templates in Secret Server. When creating a new secret, select "**Unix Account (SSH Key Rotation - No Password)**" as the Template to use this new feature.
•	Added a native Amazon (IAM key) rotation template. When creating a new secret, select "**Amazon IAM Key**" as the Template to use this new feature.

Integrations

•	Added a new biometrics eWBM integration for FIDO2 authentication with Secret Server. eWBM's website at http://www.ewbm.com/.
•	A connector application is now available to link Secret Server to identity management tools using the System for Cross-Domain Identity Management (SCIM) standard.

System Support Updates

•	Secret Server now supports Azure SQL as its database server but only if the Web server is hosted in the same geographical datacenter as the Azure SQL database (because of latency issues). Secret Server does not support Azure SQL if the Web server is hosted on-premises. **Note:** Azure SQL users are unable to use the Secret Server database backup setting. For information about Azure SQL backup please see Microsoft documentation. (https://docs.microsoft.com/en-us/azure/sql-database/sql-database-automated-backups)
•	Secret Server System Requirements have been updated to include Support for Windows Server 2019 and SQL Server 2017.

Enhancements

Session Recording

•	Updated the "View Session Data" feature for session recording to keep users from viewing session data on any ongoing session.

•	Added searching by date and time for session recording videos.
•	Updated the advanced session recording agent (ASRA) so that the SecretLauncherSessionMatchingService can now match the ShortName to the NETBIOS name of domain objects in Secret Server. * Windows terminal sessions report the pre-2000 domain NetBIOS name, which is relayed from the ASR agent to Secret Server and used for matching Windows terminal sessions to secret launcher sessions. This is an issue if the domain on the secret is the domain FQDN or friendly name and matching fails. For example: You have the ASR agent installed on a machine which is joined to a domain (mydomain.com) with a pre-2000 NetBIOS domain name "SOMETHINGELSE" Your secret has the domain name "mydomain.com" When you launch into the machine, Secret Server attempts to match "mydomain.com" (from the secret) with "SOMETHINGELSE" (the domain NetBIOS which the ASR agent relays) and fails. Secret Server resolves this issue by trying multiple methods to get the appropriate FQDN for the Windows terminal session and using the best result from those methods, which should result in matching correctly more frequently.
•	Updated advanced session recording matching by automatically mapping to the secret template's RDP fields automatically. Matching was improved by checking the secret template's RDP launcher mappings. Now, no matter what the fields are named on the secret template, the correct username and domain fields are used, based on the launcher mappings.
•	Advanced session recording can now match sessions using DNS resolution. When using an RDP launcher on a secret with a machine or computer name, instead of an IP address, the protocol handler does a DNS resolution of the hostname and reports the IP back to Secret Server, which is saved for the entry in tbSecretSession. This helps advanced session recording find additional matches when the hostname does not match the computer name, only if the hostname resolves to an IP address found directly on the target computer. If there is any NAT used, this will still fail to find a match because the NAT IP is not found on the target computer.

Secret Folder Import and Export

•	Added subfolder creation during secret importation.

Secret Templates

•	Updated launcher and password changing for the IBM iSeries secret template.

Search

•	Added default index separators to include the following characters: ? Question mark ! Exclamation point @ At symbol # Pound symbol (left parenthesis) right parenthesis [left square bracket] right square bracket { left brace } right brace ' apostrophe " double quotation mark - hyphen Configure indexing separators at **Admin > Search Indexer**.

Discovery

•	Updated the ESX discovery scanner to updated versions of the VMWare/VSphere SDK.

Logs

•	Enhanced verbose logging. Added additional logging for the Scheduled Task Scanner, Discovery Classes, Active Directory, and Password Changer logs.

Bug Fixes

Launchers

•	Fixed an issue where the Mac launcher was not automatically populating unique password prompts when connecting with PuTTY.
•	Fixed an issue where the Custom Launcher screen resolution setting for Mac Launchers was not properly applied. Previously, custom launcher sessions used full-screen resolution. **Note:** If there are no custom values set, session resolution will default to 1024x768.
•	Fixed an issue where using a Windows account launcher with a domain account would fail if the launcher was configured with an IP address in the machine field. The logic detected the computer name as an IP address and then returned before setting the domain field.
•	Updated the Mac Launcher to hide the session recording indicator for users when **Configuration > Session Recording > Hide Recording Indicator** is enabled for the launcher. The mac launcher now uses the "hideRecordingIndicator" variable stored in the sessionInfo object. Initially, there was no logic involving hiding the indicator. Added the "hideRecordingInidicators" function to the object ALaunchedTask. Updated logic under "refreshRecordingIndicator" under TaskMonitor to call "hideRecordingIndicators" when determining if it should show any icon for active watchdogs. Updated watchdog to contain a public property that is passed on instantiation–"hideWatchdogMessages". On watchdog start or stop, it now validates the "hideWatchdogMessages" is false before displaying alerts about the session being recorded.

•	Fixed a custom URL issue where launcher images did not properly display when accessing Secret Server from different sites. The custom URL was a different sub-domain from the URL that the site was accessed from. The images were served from the custom URL and our content security policy was blocking those requests. The content security policy is used to prevent cross-site scripting attacks. The solution uses the URL given instead of the custom URL.
•	Updated the Mac launcher to include support of TLS versions 1.2 and 1.3.
•	Changed the default launcher setting for RDPUseComputerForDomain to "True" to allow the launcher to work on all Windows versions currently supported by Microsoft upon install. This was tested on Windows 8.1, 10, Server 2008, Server 2012, Server 2016, as well as on MacOS 10 Sierra, High Sierra, and Mojave. It prevents launchers from failing unintentionally when connected to some Windows OS versions.
•	Updated launchers to accept upper case characters in URL strings on a Web password secret. Previously, launchers set all uppercase characters to lowercase before launching, which caused launch failures for URLs that are case sensitive.
•	Fixed an issue where a user launched a session recording but then immediately exited, Secret Server still attempted to record the session, resulting in inaccessible videos that were 00:00 seconds long. To resolve, zero-length sessions are no longer turned into videos and now returns the error: "Session too short - no video."

New User Interface

•	Fixed a bug in Internet Explorer for the new UI where the Download File button threw an "access denied" error.
•	Fixed a bug where the dependency log view did not sort correctly on the Date Recorded column.
•	Fixed a bug in the new UI where users with "edit" and "administer folders" permissions were unable to create subfolders within a parent folder.
•	Fixed a bug in the new UI where users without the "Assign Secret Policy" role permission were unable to create new secrets in a folder. The internal logic for validating whether a user may assign a secret policy to a secret did not consider the special case of secret creation. The user's personal permission to assign a secret policy should be ignored during the creation process if they did not assign one, but the code acted as if the user had assigned the policy, not that the policy was assigned by the folder.
•	Fixed a bug in the new UI where breadcrumbs for the Workflow and Teams audit pages displayed "audit" instead of the component's name. This issue also caused breadcrumbs to reload when switching between tabs on the page.

Time Conversion

•	Fixed a date-time issue where abbreviations for the month of the year in languages other than English were not properly logging to Syslog.
•	Fixed an issue where records in custom reports were logging as the UTC time zone rather than a user's local time zone.

Scripting

•	Updated REST API call "Get Folder Permissions" parameter for filter.userId to return all user permission information.
•	Added enhanced error logging for Oracle scripts.
•	Fixed a bug where the API incorrectly required Owner permissions on a secret to change the password on that secret. Edit permissions are now required when changing a password through the API.
•	Fixed an issue where the "Webservice Password Displayed" audit was logged incorrectly when non-password-related API calls to "Get Secret Fields" occurred. The REST endpoint **GET /secrets/{id}/fields/{slug}**, when used to view a non-password secret's field was incorrectly recording an audit stating "WEBSERVICE PASSWORD DISPLAYED." This was due to the endpoint's implementation using the existing **GetSecret** REST logic, which manages the creation of this audit. This logic assumed you would respond or display to the user with the secret's fields (items and data) when getting a secret that contains a password, recording an audit. To correct this, the **GetSecret** logic now supports an optional parameter that specifies which of the secret's fields (items or data) you will be responding with (displaying) to the user. If the items supplied to this method do not contain a password field, then the password displayed audit is not recorded. The corresponding **GET /secrets/{id}/fields/{slug}** endpoint was then updated to supply this method with the requested secret field, fixing this bug.
•	Fixed a bug where Web services incorrectly returned SecretDependencyTemplateIDs as null. The SecretDependencyTemplateId is now returned with every dependency found by GetDependencies in SSWebServiceHandler.
•	Fixed a bug where an "Application Account License limit has been reached" error was incorrectly displayed after editing existing application accounts, instead of only when creating new user accounts. This bug was due to a flaw in the user license logic where application account licenses were incorrectly returning the API_MetApplicationAPILimit error code when editing and attempting to save an existing application account user, while at the current application account user limit. Specifically, the logic assumed the addition of a new application account user—it did not support editing an existing user. The error occurred at the application account user limit where adding a new user is not permitted. The fix was to check if users were adding a new application account user or editing an existing user when determining whether to return the error code, API_MetApplicationAPILimit.

•	Fixed a bug where API group membership changes were not properly audited. This bug was due to a lack of auditing logic in the CreateGroupUser and DeleteGroupUser REST endpoints. The fix was to observe what audits the UI were doing upon adding and removing users to and from groups, then applying that same logic to the CreateGroupUser and DeleteGroupUser REST endpoints. Some extra logic was added to prevent recording audits if no operation was executed, such as moving a user to a group it already belonged to or removing a member from a group it was not a member of.
•	Fixed a bug where users were unable to create subfolders that inherit folder permissions through the REST API, despite the user having "Edit Folder" permission.

Authentication

•	Fixed a SAML login issue that could cause login failures if multiple users had the same User Principle Name (UPN) in Secret Server. The account that was created first would be prompted for SAML authentication even if that account was in a disabled state. This caused login failures with an error saying the user's account was disabled. Updated GetUserByUserPrincipalName to return a user only if there was exactly one user found with that UPN that was active. If there is not exactly one, it returns null. Handled a few logic flows where a GetUserByUserPrincipalName request is followed with a request to GetUserbyUsername. In cases where a valid UPN is entered, an error is produced at GetUserByUsername. These errors are now handled and logged.
•	Fixed a bug where if an instance had two users with the same username (one active user, one disabled user) across different domains two-factor authentication (2FA) enabled for login, a user trying to reset their login might get a validation error because the reset code they entered would be checked against the code of the other user with the same name. Secret Server now checks the domain of the user who is trying to login when determining which user account should be used for the 2FA reset request. This removes ambiguity when users in different domains have the same username, thus ensuring the correct 2FA validation is used.
•	Fixed an issue where 2FA failures were not showing up in the "Failed login attempts" report. The 2FA failure message was different from a normal login failure message. The report SQL was changed to display any login failure message.
•	Corrected a configuration error that changed the 'Disable Radius NAS-IP-Address Attribute' setting to have the opposite effect. The logic using the DisableRadiusNasIpAddressAttribute setting was set to return if false instead of true. This was an old application setting that would normally be false for most customers. When it was added to the UI configuration pages, the setting was implemented incorrectly, reversing the logic. In versions 10.5.14, 10.6 and 10.6.1, it is being disabled by default due to the backwards logic. This release corrects the configuration setting to work as described in the UI.
•	Fixed a bug where the authentication function for the "Attempt User Password" Radius setting did not work when using the REST API.

Discovery

•	Updated the PowerShell local account and dependency scanners to timeout after a set timeout. The application settings DiscoveryScannerTimeout and DependencySearchTimeout have been moved to settings on the individual scanners. If this setting was only set in an application setting, it needed to be reset on the scanner setting. Wrapped the PowerShell dependency and local account scanners in a System.Task to cancel after a timeout period.
•	Fixed an issue where out-of-the box dependency scanners (scheduled tasks, application pools, and service accounts) were not correctly copying as the same specified scanner's type.
•	Fixed an issue where discovery imported group-managed service accounts (gMSAs) as new computers and subsequently attempted to scan them for accounts.

Password Changing

•	Fixed an issue with Oracle privileged password changers. Before this update, Oracle password changing logic for privileged accounts was not correctly using the designated privileged secret to perform password rotations, leading to failed password change attempts.
•	Fixed an issue where missing parameters in the Sonic Wall password changer resulted in multi-factor-authentication (MFA) failure. The info sent to the Sonic Wall password changer now includes default values for MFA code and future MFA codes so those do not reference null. The PhantomJS script for Sonic Wall now looks for the host name at a different place in the parameters list.
•	Fixed a bug where the IBMi mainframe password changer timed out during process cleanup. IBM iSeries password changer has two threads during a session, the primary thread processing the session or request and a subsequent one that is spooled up during the process to ensure the interface framework (ws3270) does not get stuck in the wait state when the wait command is called. The second thread may detect false stuck wait states and attempt to close the session. If this happens after the primary session has completed the wait and has already closed, an "Access Denied" exception occurs. That exception can still be due to permissions, so it is worth checking, but most of the time this is the error thrown by the .Net Process framework when attempting to exit a process that has already exited.
•	Resolved issues with error handling, command pause handling, and session termination / logoff when using remote password changing (RPC) with IBM z/OS and IBM iSeries. IBM z/OS and IBM iSeries utilize the same base architecture within RPC to manage and execute commands. In some cases the password change process did not correctly resume after a pause command. In others the pause did not take effect, causing commands to execute prematurely and showing incorrect fails. Code was also added to clean up failed RPC / Heartbeats once a user is logged in. An identifier to the logoff command now informs RPC of the correct command to kill sessions in cases where it needs to terminate prematurely.

•	Fixed an issue after a Google update where changing Google account passwords returned incorrect messaging whether a password change failed or succeeded. Important Notes: The Google RPC only works if the user has previously logged into the Google account from the IP address that is trying to change it. In other words, the user needs to log into their Google account from a Web browser and allow the sign-in from an unrecognized IP address before attempting to rotate the password from that IP. If there are too many failed login attempts, Google may block the login for some time. If Google prompts a reCAPTCHA field to verify a human is changing the password, the RPC for the Google account will fail. Technical Details: Updated the URL check for the password changed test to fix the failed password change when it was a success. Updated the missing error code to fix the successful password change when it was failing. Also added better error checks to inform the user of the specific password change error.
•	Added logging to provide details when password change attempts for Google accounts fail, including: NewPasswordLengthTooShort NewPasswordLengthTooLong InvalidCharactersInNewPassword NewPasswordNotComplex IsAPreviousPassword SameAsCurrentPassword

Secret Import and Export

•	Fixed an issue where a user was able to edit the error message in the "Import Secrets' dialog box after importing a secret.
•	Fixed an issue where secrets were not displaying for users when a left bracket appeared in the parent folder name. We replaced any instance of [in a LIKE clause with [[] The first [opens the character comparison list. The second [is the character we allow for comparison. Finally, the closing] closes the character comparison list. It is unnecessary to replace additional instances of] because they mean nothing without a matching [
•	Fixed an exportation issue where the double quotation mark (") was not included in CSV files.

Security

•	Resolved a security issue in the protocol handler.
•	Fixed a security issue where users retained Unlimited Admin permissions for a short time after the Unlimited Admin Mode was turned off, due to a caching issue in the background worker. The issue was related to caching configuration data for 5 minutes and how the cache was being managed by the ServiceLocator used by the background worker. Turning Unlimited Admin Mode on or off would clear the configuration from the cache used by the UI, but the background worker had its own cache, which did not get cleared. The fix was to reinitialize the cache provider on the ServiceLocator whenever starting a bulk operation or secret importation to ensure these functions get a fresh copy of the configuration and Unlimited Admin setting.
•	Resolved inconsistent behaviors in role-based permissions.

Other Bug Fixes

•	Updated Session Recording to address an issue where video recordings intermittently returned black frames during the session's video playback. This was observed on older laptops running Windows 7 (either 32-bit or 64-bit) with integrated Intel graphics. Neither metadata nor keystroke logging (for advanced session recordings) were affected by this issue, only the viewing experience of the video, which could be interrupted by black frames. The fix was to use a different Windows API call as a fallback whenever a black frame is returned from the original screen capture attempt. A drawback to this approach is that fallback API simply captures the whole window including whatever might be on top of it. To counter this, logic was added to ensure that the RDP window is topmost during image capture. There remains a small possibility of capturing an isolated black frame or two if another window is moved on top of the maximized RDP window when an image is captured.
•	Fixed an issue where some Secret Server background threads attempted to perform tasks when Secret Server was disconnected from its database. The issue was that some background threads would exit their continuous monitoring loop. Specific conditions caused new SQL server connections to time out, but the open connections continued working. This resulted in Secret Server working normally but caused the monitoring loop to exit and fail to restart, breaking the monitoring of the thread. This was fixed by catching the error that caused the loop's exit, allowing the monitoring process to continue.
•	Fixed a bug where removing an event subscription item deleted the wrong item. When editing the events to trigger the event subscription, the ID of the event stored in the database identified which event to remove when an item was deleted. If an item was added, but the updated event subscription had not been saved, that ID was not generated yet. This caused the first event in the list to be deleted every time a newly added event was removed while editing. A temporary ID is now generated when a new item is added to the list so that if the item has not been saved to the database, the correct item in the list can now be identified.
•	Resolved an issue where backups were unable to complete if a Web application file was in use by a worker process. When looping through the files to be backed up, we now catch errors for any files that cannot be accessed by the backup process due to being locked by another process. When an error is caught, the issue is logged, and the backup process continues to the next file.
•	Fixed a bug in the old UI where a Safari browser repeatedly prompted Mac users to download and install the plug-in when attempting a copy-to-clipboard action. The old UI prompts for our Safari plug-in to be downloaded for using the user's clipboard. This is no longer used in the new UI. The link to the plug-in was removed. Now, the user loads the password by clicking the "load" image and then can copy the secret's value to the clipboard by clicking the copy image.
•	Fixed an issue where a Distributed Engine would return "Unknown Error" for a secret heartbeat if there was a transient error publishing the real secret heartbeat result back to Secret Server.
•	Fixed a bug where sorting groups by the "Created" field did not properly order groups in the table.
•	Improved page size restrictions when assigning roles and editing groups for customers with large numbers of users.

•	Fixed a bug where searching from a tab did not correctly output search results until the user navigated back to the Home Dashboard. To fix this bug, users now are automatically redirected to the Home tab when performing a search.
•	Fixed a bug that allowed Secret Server Professional Edition users to set a Workflow Access Request secret policy that should be unavailable in Professional Edition, causing an error when adding secrets to folders using that policy.
•	Lengthened the default maximum value for the transaction timeout on the installer from 10 to 90 minutes to prevent installs from failing due to longer database setup. The issue was caused by lengthy transaction times when running the installer against an Azure SQL database. The default maximum value for a transaction timeout is 10 minutes. We added code to allow us to make the transaction larger and set the timeout to 90 minutes.
•	Fixed a bug where converting a secret to a different template created two active copies of the secret: one converted, one unconverted. Inside the SecretDataCopier, when setting permissions, the code did not know it was due to a conversion. It then copied the folder permissions down onto the secret, instead of using the secret permissions that were on the secret originally. This meant that in the scenario above, the user did not have edit rights, throwing an "Access Denied" error.
•	Fixed an issue where launcher-based sessions were not displaying proper messaging during session recording. Message boxes are now correctly bound to their parent process (the launched process), and message box behavior has changed. Before this update, message boxes launched without a proper parent process would be minimized and would not display or block input if you restored the launched process, because they did not belong to the launched process. After this update, message boxes are launched with a proper parent process always appearing on top of that process and preventing input to that process until the message box is acknowledged.
•	Fixed a bug where Distributed Engines configured to callback to multiple Web servers did not work if the server names were separated by semicolons.
•	Fixed an issue where after an SSH key expiration occurred, uploading a new key remained in an expired state instead of properly updating the key.
•	Resolved an exception error that occurred when closing an RDPWin session window after a secret-checkout session ended.
•	Fixed an issue in customer environments configured with ticketing systems where approving a request from an event notification email caused an exception error to occur, rather than completing the approval.
•	Fixed an issue where reports containing the #CUSTOMTEXT dynamic parameter failed to send emails.

Secret Server Release Notes 10.6.000001

Release Date: 2019-03-20

Enhancements

Enhanced the Secret Server Login Assist plugin for Chrome browsers. Changed the mechanism that displays the Login Assist icon in the password field so that it is not obscured by other visible elements of the page. This update also reduced the size of the Login Assist downloaded package.

Download the updated version at https://chrome.google.com/webstore/detail/secret-server-login-assis/pbemhiacephlgacdahceanbbiokmgldb .

Bug Fixes

- Moving dependencies into groups more than once caused a JavaScript error. Fixed a bug where a JavaScript "page unresponsive" error occurred after editing many dependencies without leaving or reloading the page. To resolve this issue, we fixed a memory leak that slowed the page down and caused the error.
- Converting a single Unix Account (SSH Key Rotation) template to the same template type incorrectly populated the Private Key and Public Key File Type drop-down lists. Bulk operations doing the same thing worked fine. Fixed. Note: The customer needed the source and destination templates to be the same because they had a custom template based on it.
- Issues appeared with editing and creating administration scripts after upgrading from Secret Server 9.1.000001 to 10.5.000003. Fixed issue when upgrading from versions 9.1.000001 or earlier. Secret Server
- Fixed: Mac RDP Proxy did not work in Secret Server 10.6.
- Fixed: Custom proxied SSH process provided the wrong $HOST or $MACHINE field values.
- Fixed: An incorrect domain name appeared in the RDP launcher log on window.
- User and group "Send Email Alert for Accounts Found" drop-down did not display correctly on the Discovery Rules Alert page. Fixed an issue where the "Send Email Alert for Accounts Found" user and group "auto completion" drop-down did not display correctly. This occurred on the Discovery Rules Alert page when an SMTP server was configured for the instance.
- Fixed: Bulk 'Edit Share' option erroneously added additional users to share the secret with.
- Mac Custom launcher did not work. Fixed a bug where the Custom launcher failed to launch when configured to use the "Terminal" or "Text Edit" processes.
- Importing secrets into a folder the user owns or can edit erroneously required"Administer Folder" permission. Fixed an issue where a user without the "Administer Folder" permission was unable to import secrets into a folder that they had "Owner" or "Edit" permissions on.
- Mac Unix SSH Key launcher did not work with SSH proxy. Fixed a bug where an SSH proxy did not connect to server from a Mac launcher.
- Custom launchers incorrectly parsed the input prompt. Fixed an issue where Custom launchers (Process, PowerShell, or Batch) did not correctly parse the input prompt.
- Mac Custom proxied executable ran PuTTY instead of a process. Fixed a bug where a custom proxied launcher for Mac attempted to launch on PuTTY instead of SecureCRT.
- Mac launcher was unable to connect to some endpoints after trusting the server fingerprint. Fixed a bug where the Mac launcher was unable to connect to some endpoints due to an issue processing the target machine fingerprint.

- Session recording did not handle HSM encryption properly for video playback. Fixed a session recording issue where video decryption for playback and downloads was not handled correctly when using an HSM. Decryption was incorrectly attempting to use the HSM, rather than the master, key for decryption.
- Dependency group numbering was incorrect when adding dependencies to groups. Fixed. The dependency listing could display multiple dependencies with the same number. This could cause the incorrect impression that dependencies would be run in parallel.
- Copying a secret did not copy file attachments. Fixed an issue where file attachments were not copied when copying a secret or converting to a new secret template.
- When changing the parent folder, the API did not update the folder path for subfolders. Fixed an issue where changing the parent folder ID of a folder in the API did not update the folder path for subfolders.
- Assigning secret permissions to a group threw a secret template permissions "Bad Request" error in Secret Server 10.6. Fixed. Attempting to remove permissions from the All Vault Users group or assigning permissions to a different group threw the error. Assigning permissions to individual users still worked; however, any group-based assignment failed. Secret Server
- In Secret Server 10.6, Execute and other keywords were not filtered from custom reports, creating a security vulnerability. Fixed a security issue where some keywords were not filtered properly from custom reports. This allowed database modification by knowledgeable attackers.
- Hide Launcher Password setting did not work in all cases. Fixed an issue where the Display option was visible in the preview pane of the new UI when a user had Edit permissions on a secret, even with the Hide Launcher Password setting enabled. The password itself was never shown to users who did not have Edit permissions, so there was not a security issue.
- Web services session expired after upgrading to Secret Server 10.6. Fixed issue where the Session Timeout for Webservices setting was overridden by the Allow Remember Me setting for REST and SOAP sessions.

Secret Server Release Notes 10.6.000000

Release Date: 2/12/2019

Upgrade Notes

Manual Updates When Using Integrated Windows Authentication with Distributed Engine

Customers using the Integrated Windows Authentication (IWA) feature with a Distributed Engine need to perform a workaround when upgrading to Secret Server 10.6. IWA is the Windows feature where users log on their Windows domain only once—once logged on, any additional domain logons are done automatically without having to reenter a user name and password.

*Installing New Advanced Session Recording Agent***

Customers using Advanced Session Recording need to deploy the new agent when upgrading to Secret Server 10.6. Secret Server 10.6 does not support the RDP Monitoring Agent from Privilege Manager for recording keystrokes or process auditing.

New Features

Workflows

- Added a new enterprise feature to allow multi-tiered secret approvals. Workflow Administrators can setup secret access to be granted to users only after meeting multiple layers of approver requirements. This feature is available in Secret Server's new UI only.
- Added two new Roles to support multi-tiered secret approval workflows:
- Workflow Administrator – Can administer workflow permissions.
- Workflow Designer – Can create new workflow templates.

Advanced Session Recording

- Enhanced speed and performance for both Basic Session Recording and Advanced Session Recording.

  Added a new agent to Secret Server for Advanced Session Recording that captures metadata from launcher sessions to targets.

  Customers using Advanced Session Recording need to deploy the new agent when upgrading to Secret Server 10.6. Secret Server 10.6 does not support the RDP Monitoring Agent from Privilege Manager for recording keystroke or process start data.

  New Advanced Session Recording Agent includes functionality to uninstall themselves when deactivated from Secret Server.
- Added new configuration setting for storing session recordings based on site. Navigate to **Admin > Configuration > Session Recording tab > "Enable Moving to Disk" > "Archive Location Dependent on Site."**

  Note: "Enable Moving to Disk" must be checked to reveal the new setting. You may then specify folder path locations based on each site that the Session Recording secret uses.

  Secret Server must have appropriate permissions required to access all file paths listed for saving the configuration and recordings.

  When creating a new site, a specific site-to-folder-path relationship won't be created automatically, the secret will instead use the default path (i.e. whatever path is used for the local site).

  When you edit, a row will show up with the default value per site already loaded.
- Added a warning for Session Recording experiencing many unprocessed videos.

Teams

- Added a new "Teams" feature so that Secret Server administrators can segment users and groups within one Secret Server instance.

  Designed as a convenience feature, users that lack an elevated role permission within a team will not be able to select users, groups, or sites that exists outside their team, including in dropdown options and searches.

  Added three new roles to support teams:

- Administer Teams - Can create, edit, and view teams.
- Unrestricted by Teams - Can view all users, groups, and sites. The default User role in Secret Server now has this permission, so role permission customization is needed for Teams to take effect.
- View Teams - Can view all teams.

FIDO2 (YubiKey) Authentication

- Added a new integration with Yubico. Secret Server can now be configured to use FIDO2 tokens (YubiKeys) as a method for multi-factor authentication.

Launcher Compatibility

- Added backwards and forwards compatibility to the Secret Server Launcher Protocol Handler.

Telemetry

- Added a new feature that sends anonymized usage data to help guide future research and development plans at Delinea.

Remote Password Changing

- Added the ability to use Secret Server's API for Remote Password Changing on AWS secrets. That is, support for PowerShell script password and dependency changes for AWS IAM token rotation. Generated values are passed back for saving on the secret. This useful for tokens generated by an external system during rotation. **Note:** this is only the underlying architecture for use via PowerShell scripts, not a full password changer.
- Added a new IBM iSeries Password Changer template to enhance 5280/IBM Series 7.1-7.3 Systems support including additional features like program functions.
- Added the ability to use a $$Pause Command for the custom SSH Password Changer so that administrators have the option to prevent run commands executing immediately after login, which can cause failed executions.
- Added support for VMware Password Changing and Discovery to work with updated versions of PowerCLI up to 10.1.1.
- Increased default RPC Retry Interval to run every 15 minutes and to cap at 10000 consecutive tries. "Unlimited max attempts" is no longer an available option. Retry Interval can be manually configured by 5-minute increments. Heartbeat interval is also now capped at minimum of 15 minutes.

SDK

- New version of the SDK released on nuget.org that supports targeting for .NET Framework 4.5.1 in all packages. The only external dependency is Newtonsoft.Json (v11.0.2). https://www.nuget.org/packages/Thycotic.SecretServer.Sdk

RabbitMQ Helper

- Added Federation Support and Clustering functionality for RabbitMQ Helper.
- Federation: https://thycotic.github.io/rabbitmq-helper/usecases/federation/
- Clustering: https://thycotic.github.io/rabbitmq-helper/usecases/clustering/

Enhancements

- Added the ability to download a recorded session from the API.

  This new API call will not download metadata, only applies to Basic Session Recordings. The call is api/v1/recorded-sessions/{id}/session-recordings.
- Added a PrefetchCount app setting to allow customization of engine response message processing and enhance processing speed.
- Enhanced load performance for Discovery.

  Added a cache for Session Configuration to prevent excessive callbacks to SessionRecordingConfigurationProvider.Load during Discovery scanning.
- Added a new setting for configuring the SSH Remote Password Changing timeout interval to all applicable SSH Secret Template Settings' pages.

  Prior to this fix, users were not notified when a group name exceeded the maximum character length and instead experienced a web session hang.
- Added a setting to Discovery that can check whether IIS is installed before scanning for Application Pools. The "Verify IIS is Installed" Setting can be enabled by navigating to **Admin > Discovery > Edit Discovery Sources > [Select a source] > Scanner Settings >** Scroll down to under the "**Find Dependencies**" section and click the **edit icon** next to the **"Application Pool" Scanner > "Settings – Application Pool"** page.

  In Extensible Discovery, scanners are setup to run in consecutive order across many organizational units within large environments. This setting was added to allow Secret Server to skip the process of scanning for application pools whenever a machine does not have IIS already installed to enhance performance and cut down on run times.
- Added support for a RADIUS challenge in web services. Secret Server will now return a "RadiusUSAccessChallenge" error if an additional prompt is needed. To use this functionality, on-premises Secret Server needs to connect to the same node on both authentication calls.

  Previously, Secret Server could only handle a single request from RADIUS' authentication process. This enhancement uses caching, which means that authenticating scripts need to hit the same web node to use a challenge authentication. This fails when using REST + Secret Server On-Prem + Load Balancing + RADIUS Challenge Authentication combined. The workaround is to hardcode REST scripts by IP or FQDN.
- Added various enhancements to the Upgrader. Enhancements will not take effect until your next upgrading process.

  Upgrade enhancements include updated logging, the removal of unchanged files during the upgrade process, re-ordering of tasks to improve performance, and enhanced messaging.
- Enhanced error messaging when Heartbeats fail due to an unavailable machine state.

  Prior to this enhancement, when Heartbeats failed on a machine due to disconnection it flagged an "Unknown Error." Now the machine will return an "Unable to Connect" status.
- Added and enhanced the Custom SSH Password Changer's console output logging for debugging.

  Prior to this issue, command set logging was removed from Custom SSH Password Changers due to a security concern in the logging messages. To resolve this issue the original security messaging was addressed, then debug logging was reinstated.

- Fixed an issue where Heartbeat failed when changing a AS/400 Mainframe secret due to versioning requirements.

  When setting up an AS400 Password Changer for V7R1 or V7R2 systems, the page erred due to a timeout response from the password changing that failed to manage the WS3270 utility properly. Updated the supported versioning for the IBM iSeries Password Changer template to resolve this issue.

- Added the option to include subfolders when configuring event subscriptions.

  Users can now optionally apply event subscriptions to subfolders, previously folder subscriptions were limited to individual folders.

- Enhanced performance of the Event Subscriptions' page.

  Page performance on the event subscription page in Secret Server was increased by rearranging the logic and ordering of the event subscription processes.

- Increased default LDAP processing throttling limit on Distributed Engine from 100 to 1000.

  Distributed Engine's performance was slowed down by the low throttling threshold. Increasing the limit allows faster performance for Engine tasks like heartbeat or password changing.

- Enhanced audit logging messages when viewing and displaying secrets and Passwords.

  Unmasking a password or viewing a password's History is only logged in Secret Server every five minutes when the same user performs the same action on a secret within a short period of time. Prior to this update the description of behavior on the Audit View page did not include the action for "Password Displayed."

- Two-Factor Authentication now supports the User Principal Name (UPN).

  Two-Factor Authentication now supports using the User Principal Name (UPN) as a default for usernames when logging into Secret Server. Configure this setting in **Admin > Configuration > Login > RADIUS Default Username** when RADIUS is enabled.

- Renamed the "Google Authenticator" dropdown option for Multi-Factor Authentication to "TOTP Authenticator" (Time-based One-time Password algorithm) for accuracy.

- Updated Distributed Engine to use REST instead of Windows Communication Foundation (WCF) when using HTTP to contact Secret Server. WCF is no longer a prerequisite installing Distributed Engines.

- Added log message buffering.

  By default, log messages are not buffered and are written to logs immediately. For very active systems, log message buffering can increase performance. In the web-log4net.config file, the Delinea.BufferingForwardingAppender parameter should be uncommented and then IIS reset. Buffer size can be configured using the bufferSize parameter in that configuration file.

Bug Fixes

- Fixed a bug where manually added fields in custom templates would not display in the table view.

  Prior to this fix, when creating a custom template with the "expose for display" option selected the field would not display in the table as a new column.

- Corrected error messaging when deleting a dependency on secrets with large numbers of dependencies.

When a secret had 1000+ dependencies, deleting one of them resulted in a "Failed to Execute" error prior to this fix.

- Fixed error reporting when a user enters a group name that exceeds the max character length.
- Fixed an issue where background threads could go into wait mode indefinitely during a manual upgrade.

  Added detection for when the "upgrade in progress" indicator remains active after a completed manual upgrade. This could cause background tasks to remain paused indefinitely during a manual upgrade. Now, the user is notified and can reset the indicator, allowing background tasks to resume.
- Resolved an issue with Amazon secret heartbeats and rotations because of a change in Amazon login page layout and flow. To create a new Amazon Web Services (AWS) secret: create a new web password secret, then select **Remote Password Changing tab > Password Type** (Amazon, Google, and Salesforce options).
- Fixed an issue where a RADIUS Authentication timeout would block other RADIUS requests from authenticating to Secret Server, causing login delays for RADIUS users.

  Prior to this update there was a lock in the RadiusUSRequest.GetResponse() that only allowed one connection at a time. This fixes an issue when making more than one UDP connection to the same client port.
- Removed unnecessary logging on the OperationCanceledException in the System Log.
- Fixed a bug where SSH Key rotation commands were not properly authenticating.

  The "Verify" command for password changing was ignored if no "Post Change" command existed in a Custom SSH key rotation template, resulting in being able to use the same key to connect for the verify command and the password change in certain cases.
- Fixed a broken documentation link on the Server Nodes page.
- Fixed a bug where duplication errors occurred if scheduling a report with the same name as another, already disabled, report in Custom Reports.

  The workaround for this issue was to undelete the original report and rename it before deleting it, but users are now able to delete and create reports with the same name, if the duplicated reports are not both active.
- Fixed an issue that prevented users from creating or editing an SSH Command Menu on individual secrets.
- Fixed an issue where the "Allowed" and "Available" Secret Templates' columns did not populate on the "Create New Folder" page when restricting secret templates.
- Fixed a bug where editing checkout hooks saved changes instead of allowing users to cancel out of the edit page.

  After clicking cancel on the checkout hooks edit page, the updated data was not saved to the database but was updated on the page behind the UI modal.
- Edited PUT /secrets/{id}/fields/{slug} parameters in API documentation to use Secret ID instead of the secret name.
- Edited /secrets/{id}/check-in REST API call in API documentation.

  In this script, "Force Checkin" could bypass the Access responses, then successfully check-in the secret, before attempting to return the access responses it previously skipped. This failed.

- Fixed a bug where ProcessShouldScanComputers resulted in collation conflicts after running Discovery on OUs with incorrect server collation settings.

  Specifically, this occurred if a server's default collation was set to "Latin1_General_CI_AS" because it conflicts with "Latin1_General_CI_AS" collation on Secret Server SQL database's collation settings when running SQL commands on specific tables.

- Fixed an issue where ExpiredSecretMonitor stopped running in certain conditions due to a session recording call.

  Users were experiencing several days in between these issues. The SessionRecordingBlobWriter registered an exception error that stopped background threads from processing. This SQL call was irrelevant, and its elimination allowed the ExpiredSecretMonitor to update every minute as normal.

- Fixed an issue where "Language Resource Not Found" errors were thrown on the Themes page under the Advanced section.

This issue resulted from missing resource strings on the page.

- Fixed an issue where the SSMS launcher did not send the correct password if a caret symbol (^) was in the password.

  This issue was specific to SQL Server Management Studio (SSMS.exe). The fix for this involved updating the SSMS process launcher configuration to allow for an optional escape character to be added, now configurable in the advanced section when setting up a launcher for a SQL Server Account secret.

- Fixed a bug where failed password changes on custom SSH secrets would not stop processing when CheckContains command failed.

  This issue stemmed from the CheckContains command script in the Custom SSH password changer.

- Fixed a bug where environments using geo-replication threw errors when attempting to re-send failed syslog messages.

  Prior to this fix, geo-replication environments with failed Syslog messages resulted in large numbers of error messages because Syslog created triggers on the tables and would try to update the items repeatedly.

- Fixed an issue where items could be imported into the root personal folder.

  Prior to this fix, administrators were able to migrate secrets into the Personal Folder (root) using the upload XML tool, which allowed all users to see the imported items.

- Fixed an issue where a language setting caused errors.

  Some language settings could cause an IIS crash with HTTP Error 503: The Service is Unavailable.

- Edited tooltip wording for the advanced "Auto Change Schedule Interval" setting.

- Fixed an issue where SSH key rotations were not properly rotating and then deleting the old SSH key at endpoint when the SSH Custom Password Changer was configured in a specific way.

  Key rotations for Linux SSH keys failed to cleanup old SSH keys on target machines after a key rotation occurred due to a missing command in the success script.

- Fixed an issue where users were unable to delete Reports.

  Deleting and undeleting Reports threw an exception error due to a database ReportCategoryId mismatch.

- Fixed an issue where the API call "GET /secret-templates" did not support the inactive filter ("filter.includeinactive").
- Updated API Rest documentation to not label a Dictionary object (enumerated KeyValuePair) as an object[].
- Fixed an issue where scheduling Reports failed if the time zone was set to time zones ahead of UTC.

  Prior to this fix, any time zone that was UTC+x failed to properly populate ScheduleCustomReportEdit.aspx "Recurrence Scheduled Start Time."
- Fixed an issue with the API Folder Create method where folders created through the API would error when trying to access permissions.

  After creating a new folder via the REST API, a Get command for the new folder would return null permissions prior to this fix.
- Fixed a bug where Basic Users were not able to use the "Create New" secrets template when Active Directory template permissions were revoked.

  In Basic User mode the "Create New" secret templates drop down list incorrectly started with `<Active Directory>`, causing permissions errors when the Active Directory Template was revoked for Basic users. The drop-down list now starts with `<Select>` in all user modes.
- Fixed a bug where searching for a subfolder would not return results when the user did not have permissions to view its parent folder.
- Fixed a bug where the bulk operation mode did not move multiple secrets to a folder.

  This issue specifically involved the search feature on the bulk operation dropdown when filtering for the folder location. Users were required to manually select the folder rather than being able to find it through search.
- Fixed issue where "Require Approval Access" on a secret policy did not follow default settings to not enforce the policy.
- Fixed a bug where a user's display name could be left null when creating new users.
- Fixed an issue where Discovery Computer Scans using a "machine only" resolution type resulted in an exception error rather than completing the scan.

  A null exception was thrown when scanning for machine names rather than using the "Use Fully qualified name (recommended)" setting in Discovery.
- Enhanced error messaging when running Dependency scans in Discovery.
- Error message fixes included:
- If all scanners worked but did not find anything: "No Dependencies."
- If one scanner fails, then it shows a failure message for the scanner.
- If multiple scanners fail, it only has room to show one failure.
- If a scan fails but no failure message exists, it will show "Unknown Error."
- Updated two collation settings in session recording tables to allow case insensitivity on table names and prevent collation errors when the SQL server collation mismatches.

- Fixed an issue from a patch update where scanning for scheduled tasks in Discovery failed due to an auto-appended domain in the authentication string.

  This issue only affected customers who deployed the December Patch, v 10.5.14.
- Fixed a bug where the "Only change password when Secret is expired" checkbox would not save if Auto-Change Schedule was set to None.
- Fixed an issue where SAML authentication check could not be disabled.

  Prior to this fix, some accounts would receive an 'Invalid Relay State' error when logging in. This had to do with disabling the "SAML authn context" in Secret Server.
- Enhanced error messaging when Heartbeat is not assigned to a PowerShell Script Password Changer.
- Fixed a bug where log file exports were not downloading from the Manage Secret Access Request page.
- Fixed a bug where the "Use Custom Window Size" Launcher setting was not implementing the correct resolution.

  Prior to this fix, when attempting to launch at a smaller 1024 x 768 resolution, a session launched instead as full screen.
- Fixed an issue where if the Response Bus Site Connector disconnected, it prevented the web site from loading.

  Updates were made to BackgroundScheduler, BackgroundWorker, and the EngineWorker. If a role cannot connect to its response bus an error will now be logged and the site will still load.
- Fixed a bug where Heartbeat on local administrator accounts on Windows Server 2012 pre-R2 was not compatible with PowerShell v3.0.

  Added output logging to file and operating system information.
- Fixed a bug where reports were saved in UTC time rather than in local server time.
- Fixed a bug where API calls against restricted secrets threw an object reference error.

  This fix allows accessing "require comment," checkout, or other restricted secrets through the API. These secrets were returning errors due to a null reference linked to a "ViewTracker" secret attribute.
- Fixed an issue where a WinRM warning message displayed even when WinRM is running and correctly configured on a server if authentication account did not have access to running services.

  Prior to this fix the WinRM warning message saying the service isn't running displayed on the PowerShell maintenance window even though WinRM was running and correctly configured on the server according to instructions at: http://support.thycotic.com/KB/a303/configuring-winrm-for-powershell.aspx PowerShell scripts ran successfully, but the misleading message remained.
- Fixed an issue where Web Password secrets caused Bookmarklet JQuery Exceptions when the Web Password Filler was used in the IE11 browser.
- Fixed a bug where EnableFrameBlocking was not respected on certain pages in Secret Server.

  Middleware was added to send X-Frame-Options / SameOrigin header for everything but the bookmarklet if not installed, no one was logged in, or FrameBlocking was enabled. MainLayout also changed for MVC pages so that they will render with the frame blocking script.
- Fixed a bug where email configurations were not properly saving applied settings.

Prior to this fix, changes to Email configuration page did not properly update the database.

- Fixed workflow issue where users were directed to the Setup Home page after saving changes on the Email configuration page.
- Fixed an issue where AD users were not receiving approval emails after an access request for a secret in some environments.
- Fixed a bug which caused audit tables to display date and time in UTC rather than the configured time zone.
- Fixed a bug that could cause bulk operations to fail in environments using geo-replication.
- Fixed a bug that could cause a "Failed to load history" error when viewing the history data for a secret.
- Fixed a timeout on secret audit when there are very many audit entries.

  Audits with very large result sets only display the top 1000 entries.
- Resolved an error reporting issue with Unix Account (SSH) and Unix Account Custom (SSH) account types so that connection failures and login failures are correctly reported.

  Previously, Unix Account (SSH) would report login failure as UnknownError and Unix Account Custom (SSH) would report login failure as UnableToConnect.

Secret Server Release Notes 10.5.000003

Release Date: 9/24/2018

Enhancements

Session Recording

- Updated Session Recording to support H.264/MP4 video encoding with resolutions up to 4K (now the default).

  Customers are now able to configure the time when Session Recordings are moved from the database to disk to suit business needs. Previously this move was hard coded to run at 02:00 UTC. Time is configured through an AppSetting.

User Interface

- Preview Mode for the new and improved redesign of Secret Server has been enabled. Please contact UX@thycotic.com to gain access to the Preview Mode.
- Included a "24 hours" option to the expiration Quick Picker.

APIs

- Updated the UserCreateArgs to include AD Guid to account for possible duplicate users in Active Directory through the API.
- Added API call for Check Out Secret.

Time Zone Enhancements

- Enhanced time zone functionality on the backend to account for daylight savings time.

Bug Fixes

- Fixed a bug where users with the "Administer Configuration" Role Permission could see but not access option settings under Admin > Configuration.
- Fixed a bug where the Application Pool scanner and IIS tester do not properly initialize the ManagementScope for WMI calls to Local Machines.
- Fixed a bug where only one Discovery Scanner worked when Active Directory had more than one local account scanner setup.
- Fixed a bug where valid REST/SOAP API tokens did not permit login access.
- Fixed an issue that allowed personal folders to be moved to the top level of the folder tree.
- Fixed an issue where after adding and saving an identity provider in advance settings (SAML), the user observed an error stating 'Object reference not set to an instance of an object.'
- Fixed a bug where a user was unable to delete an approved user from the Edit Secret Policy page.
- Fixed a bug where there were inconsistencies in the time zone handling.
- Fixed a bug where there was an error message of 'Connection Failed' during Custom Unix (SSH) Remote Password Changing but the password changed successfully.
- Fixed a configuration issue with the SSH Key Rotation Privileged Account Password Changer that is used by the Unix Account (Privileged Account SSH Key Rotation) template and caused RPC to fail with an error saying Associated Secret is required.

Security Fixes

- Fixed an issue with Request Access when set to be required for Owners and Approvers where an Approver could approve their own request through JavaScript manipulation. Note an email detailing the approval would have been sent to all other approvers.
- Fixed a Security Vulnerability that allowed Cross-Site Scripting.

Secret Server Release Notes 10.5.000001

Release Notes 10.5.000001

Release Date: 8/15/2018

Security Fixes

- Secret data is now authenticated with HMAC-SHA-256. This addresses CVE-2018-18002.
  - Security Issue Discovered by: Jarrod Farncomb of TSS (https://dtss.com.au)
- SOAP API updated to use the same token generation as the REST API.
- Security Issue Discovered by: TSS (https://dtss.com.au)

Enhancements

- Customers are now able to configure the time when session recordings are moved from the database to disk.

Bug Fixes

- Fixed issue where after 10.5 upgrade, session recording could not be configured to use a file share.
- Fixed an issue where there was an intermittent first login fail on fresh install.
- Fixed issue where selecting "Now" for a password change in Autochange Schedule could set the incorrect time in situations where the user's computer is in a different time zone than the Secret Server host.
- Fixed issue where dependency scanning would miss Scheduled Task dependencies in situations where two domains with the same name (ex: customer.com and customer.net) were used.
- Fixed issue where the password for an account would be changed multiple times in a row if the "Only Change password when Secret is expired" box was selected in Autochange Schedule.
- Fixed issue where bulk operations could cause database deadlocks.
- Fixed issue where Distributed Engine callbacks could cause a race condition.
- Added a warning to the schedule page that specified the time will be saved is Secret Server configuration time.
- Fixed an issue where after adding and saving an identity provider in advance settings (SAML), the user observed an error stating 'Object reference not set to an instance of an object.'

Secret Server Release Notes 10.5.000000

Release Date: 7/10/2018

Enhancements

SAML

- Added a new SAML configuration page so SSO providers can be configured without modifying the saml.config file.

UI Updates

- Subfolders can now be created within a user's Personal Folder.
- The login screen domain dropdown menu can now be disabled for customers that wish to hide the list of domains.
- Added auditing for the following configuration changes: Character Sets, Password Requirements, Event Subscriptions, Role names, Backups, Custom Password Changers, Licenses, and Database Connections.
- Added export functionality for Heartbeat logs, Remote Password Changing logs, Discovery logs, and Computer Scan logs.
- Failed password changes now display an error within the Secret View UI and a link to a Remote Password Changing Errors KB article.
- If using multiple devices per user in Duo Security for two factor logins, Secret Server will now show the Device Name set in the Duo admin portal next to each device.

Discovery

- Added Discovery settings to scan for open ports on target machines, connect to specific ports, and set a default timeout for port scanning.

- Added a Discovery scanner setting for excluding services, tasks, or application pools by name.
- Added new Diagnostic logs to address duplicate Discovery Scan items.

Password Changing

- Auto Change Schedules can now be configured so that they will trigger a password change even if the Secret is not expired.
- Added the ability to rotate SSH Keys with no passphrase required.

Reporting

- Added Session Revocations to the User Audit report.
- Added IP addresses to the login failure report.
- Added new chart options for custom SQL reports.
- Added the ability to configure AS400 Password Changers.
- Enhanced Secret Server's ability to process larger message sizes. Secret Server'
- Distributed Engines now send operation results back to Secret Server through the Site Connector instead of sending them via website.
- Ticket System Integration can now be configured to work over Distributed Engine.
- Improved System Log Searching in active environments.
- Added new Syslog event messages for SIEM integration and enhanced log messaging.

Security Enhancements

- Added a configuration setting to change the default role that a new user receives.
- Added the ability to audit certificate verification errors for Active Directory calls over LDAPS and syslog connections using Secure TCP.
- Added the ability to send a client certificate with Active Directory calls over LDAPS or syslog connection using Secure TCP.

Bug Fixes

- Fixed issue where Secret Policy changes would not apply to all Secrets.
- Fixed issue where Service Account Discovery could timeout and flag Secret Dependencies for removal.
- Fixed issue where Two Factor could prevent a "Login Failed" audit on the user; added new logging details in the Audit User Log if errors do occur from Two Factor authentication.
- Fixed issue where excluding OUs from Discovery scans prevented computers from being deleted when they were removed from AD.
- Fixed condition in certain environments where the auto-change Secrets were not changing properly.
- Improved performance of the Discovery Stored Procedure for specific OUs scanning to avoid timeout in large environments.
- Fixed a logging exception in Monitor Logging.

- Resolved a permission error in certain environments that occurred during Local Account Discovery Scans.
- Fixed issue where integrated Windows login requests were building up in the tbOauthExpiration table.
- Fixed issue where columns could not be sorted in Discovery Network View.
- Fixed issue where queued RabbitMQ messages were lost if RabbitMQ was restarted.
- Fixed bug where an email config port change was logging the new port as the old port.
- Fixed an issue with SQL Replication where indexes on indexed views were not replicated.
- Fixed issue where a DependencyResolutionException could occur on the Login page and prevent use of site until an IIS reset was performed.
- Fixed issue with SSH password rotation/Heartbeat connections that were reporting "Unexpectedly inactive."
- Fixed issue where the Secret Server Clipboard Utility could not be installed with Chrome 67.
- Fixed issue where the dashboard Add Content dropdown displayed below the Secrets table.
- Fixed null reference bug that occurred when autocomplete textboxes were used in lieu of a dropdown for the Group/User selector.
- Fixed bug in Password Changing where a large number of Secrets targeting the same resource for certain password changers could prevent processing.
- Fixed issue where root folders could be created through the CSV import process while that user did not have the Create Root Folders role permission.
- Fixed an issue that could cause occasional black flickering to appear in Session Recording videos.
- Fixed issue where users could be logged out of Secret Server due to inactivity while actively browsing certain pages.
- Fixed issue where new Secrets displayed the option to save to a user's Personal Folder even if Personal Folders were disabled.
- Fixed issue where Discovery local account scans were parsing unnecessary data and taking more time than necessary in large Active Directory domains
- Fixed issue where changes to Users would not save when extremely large numbers of AD groups were being synchronized.
- Fixed issue with custom PowerShell Ticket System integrations where entering a ticket number to view a Secret would produce an error.
- Fixed issue where creating a new Event Subscription would fail when specifying a user or group.
- Fixed issue where the REST API would not correctly implement the "Require Two Factor for Web Services" configuration option.
- Fixed issue where Heartbeat could be stuck in Pending status.
- Fixed issue where creating new Folders would fail when there were no other Folders.
- Fixed condition in certain environments where the auto-change Secrets were not changing properly.
- Fixed issue where Dashboard searching was slow in environments with large numbers of Secrets.
- Use of SAML while Virtual Assist Keyboard is disabled no longer causes Virtual Assist Keyboard to appear.

Security Fixes

- Fixed issue with Unlimited Admin permissions and managing Groups.
- Deprecated TLS 1.0 in Security Hardening Check.
- Fixed issue with script names.
- Fixed issue with Upgrade Status log.

Secret Server Release Notes 10.4.0

Release Date: 1/17/2018

Note: Customers with trial licenses in their Secret Server instance who upgrade to 10.4 will be required to activate those licenses.

Enhancements

- Secret Server SDK: The Secret Server SDK replaces and improves upon the existing functionality of the Java API and .NET/Application API. Users can leverage this SDK to tokenize credentials in scripts and configuration files for .NET web applications. The SDK can also call for a REST Web Services authentication token for added functionality. Finally, the SDK has a local encrypted cache for every location it is installed in to allow for quicker transit times and resiliency in case communication with Secret Server is lost.
- Session Monitoring Multiple Nodes: Clustered Secret Server environments will now automatically split the processing load for Session Monitoring events.
- AD Credential Cache: Secret Server can create an encrypted storage for hashes of Active Directory users' credentials that is updated on every successful authentication to Secret Server and allows for users to continue to access Secret Server even if communications are lost between Distributed Engine and the Domain Controller (s).
- Secret Server now has a REST endpoint to manage Secret dependencies.
- Secret Server now has a REST endpoint to manage groups.
- Secret Server's Firefox browser extension has been updated so it can continue to work with the latest versions of Firefox.
- Secret Server can now communicate to SIEM tools using a TLS connection.
- TLS connection successes and failures with Active Directory Synchronization and SIEM integration are now audited.
- Secret Server can now send audits to Windows Event Logs.
- Enhanced Dashboard search performance.
- Enhanced Active Directory synchronization performance.
- File uploads on Secrets now have file extension and size restrictions.
- Added a Security Hardening report to indicate whether email communications are set to use HTTPS.

Bug Fixes

- Fixed issue where SSH custom password changers failed Heartbeat when using SSH Key authentication.
- Fixed issue where Discovery would incorrectly show as misconfigured despite having configured Discovery sources.
- Fixed issue where random password generation may generate a password containing 2 consecutive characters contained in the Active Directory username.
- Fixed issue where the login assist for Chrome would not work if an IP address was in the URL field of a Secret.
- Fixed issue where numerous domains in Discovery could cause failures in communicating with the SQL database.
- Fixed issue where Discovery would only use the correct Distributed Engine Site on the first dependency type when scanning for all dependencies across multiple domains.
- Fixed issue where a significant number of Secrets that were failing a password change would freeze Distributed Engine.

  Security Fixes
- Fixed XXE issue on Web Launcher configuration page.
- Fixed XXS issues on New Engines management page.
- Fixed XSS issue on Secret Dependency Changers page.
- Fixed issue where Secret Server upgrade logs written to the server hosting Secret Server were accessible via a URL.
- Fixed issue where Remember Me for 2FA bypassed Duo's deny logon security settings on users in some cases.
- Updated PuTTY client for Session Launching to v0.70.
- Fixed potential security issue where the RDP service port on the client machine could be used to open a concurrent RDP session to the target machine when RDP connections were proxied through Secret Server.
- Fixed LESS Injection issue on Theme management page.
- Fixed issue where language resource ASHX files in Secret Server could be accessed by any user with access to the application.
- Fixed issue where default settings on PuTTY logging settings could expose the password of a user.

Secret Server Release Notes 10.3.x

Release Notes 10.3.000015

Release Date: 9/25/2017

Enhancements

- Secret Server now supports SafeNet Luna Network HSM 7.

Bug Fixes

- Fixed issue where Active Directory Synchronization may become slow or unresponsive when synchronizing a significant number of Active Directory groups.
- Fixed issue where users may be removed from a group in Secret Server if that group contains users from multiple Active Directory domains and one of the domains cannot be reached during a synchronization. Secret Server
- Fixed issue where Secret Server Free customers could not manually add Active Directory users.
- Fixed issue where selecting specific Organizational Units for an existing Active Directory discovery source may not work properly after a manual host range is added to it.
- Fixed issue where Unix machines discovered using PowerShell may reflect an incorrect Organizational Unit.
- Fixed issue where users were no longer forced to change their password upon first login if the Enable Local User Password Expiration configuration setting was enabled.
- Fixed issue where using Discovery for dependencies over Distributed Engine may return incorrect results.
- Fixed issue where multiple Discovery host range scans could cause redundant machine loads.
- Fixed issue where using search in Secret Server while on the Notifications page may cause an application error. Secret Server
- Fixed issue where users who have an externally facing IP address that could not be resolved from the web server hosting Secret Server may experience performance issues throughout Secret Server.

Security Fixes

- Fixed XSS issue on the Secret Template Permissions page.
- RabbitMQ Helper has been updated to install RabbitMQ 3.6.12 which now supports Erlang 20. This patches Erlang 18's vulnerability CVE-2016-10253. We do not believe that Erlang 18's vulnerability has a direct impact on Secret Server's use of RabbitMQ, but we recommend updating current Erlang and RabbitMQ deployments to these versions to keep systems patched.
- Fixed potential security issue with Secret Server's scripting functionality.

Release Notes 10.3.000014

Release Date: 8/29/2017

Enhancements

Secret Template Edit Launcher Configuration Enhancements for PuTTY Launchers $1 $2

- Added a report in Secret Server to display what Secret Template permissions a user or group has.
- Added an option in Secret Server to backup Privilege Manager.
- Secret Access Requests can now be found under the Tools menu.
- "Share Secret" Role permission has been renamed to "Own Secret".

- Upgraded the module responsible for Office 365 and Azure Active Directory password changes to ensure continued support.

Bug Fixes

- Fixed an issue where pages involving groups could not be saved if there were 5,000 groups or greater in Secret Server.
- Fixed an issue where Active Directory users in child domains may not be properly disabled in Secret Server when they are disabled or removed from Active Directory.
- Fixed an issue where ambiguous errors were logged when the username or password is correct on any password changers, dependencies, or scripts using SSH.
- Fixed an issue where the Share button would disappear on Secrets on the dashboard when a user does not have the Share Secret Role permission. Users should still be allowed to view the permissions on a Secret even if they cannot decide who that Secret is shared with.

Security Fixes

- Fixed potential security issue where a formulae injection could occur on exports from Secret Server.

Release Notes 10.3.000000

Release Date: 7/12/2017

Enhancements

SSH Key Management Enhancement

- SSH Keys as Dependencies (May require an additional license): Multiple Public keys that reference a single Private key can now be stored as dependencies on the Private key Secret. For more information, please see the Remote Password Changing section of the "Secret Server Documentation" on page 1.

EMEA Cloud

- Secret Server Cloud is now hosted out of Germany as well as the US.

Secret Template Granular Enhancements

- Added ability to restrict Secret creation for users by Secret Template.
- Added ability to set the allowed Secret Templates for a folder.
- For more information, please see the Folders and Secret Templates sections of the "Secret Server Documentation" on page 1.

SAML

- SAML Single Logout is now supported.

UI Updates

- The Dashboard load time performance has been optimized.
- Add auditing for configuration changes.

- Added a button to export all logs under Admin | Diagnostics.
- Refined user experience around creating dependencies on Secrets.
- Added bulk operations for Dependencies.

REST

- Added endpoint for IP Address Restrictions.
- Added endpoint for generating a password.
- For more information on these REST endpoints, please see the 10.3 REST API Guide.

WARNING: Customers who are upgrading from 10.2.000018 and use Windows Integration Authentication to the database may see an error message titled "Login failed for user" during the validation step of the upgrade process. To bypass this error, please use the legacy installer by navigating to `[Your Secret Server URL]/installer.aspx?patch=true&useLegacyInstaller=true`

Bug Fixes

- Fixed issue where Local Account Discovery used unnecessary calls.
- Fixed issue where Secret Server removed dependencies that were not found by Discovery and would not re-add them if found again.
- Fixed issue where the Delinea RDP Launcher would not allow fullscreen mode.
- Fixed issue where Discovery did not properly detected Scheduled Tasks on Windows 10 machines.
- Fixed issue where testing a SSH script in Secret Server would only display an exit status when it failed.

Security Fixes

- Fixed potential security issue with multi-line files and Secret fields.

Secret Server Release Notes 10.2.x

Release Notes 10.2.000019

Release Date: 6/19/2017

Enhancements

- Privileged accounts assigned on a Secret Template now take precedence over privileged accounts assigned on Secret Policy.
- Secret settings are now able to be modified via the SOAP web services API when a Secret is checked out.

Bug Fixes

- Fixed issue where Secrets with privileged accounts assigned for password changing could not be moved to a folder with a Secret Policy that contained a privileged account.
- Fixed issue where Secret Server could delete recordings stored on a disk.

- Fixed issue where users without the View Deleted Secrets role permission received errors when expanding the advanced search bar on dashboard.
- Fixed issue where some accounts discovered by Discovery were not matched to existing Secrets.
- Fixed issue where upgrade may fail when the database connection is configured to use Windows Authentication.
- Fixed issue where "Show Proxy Credentials" on a Secret will fail when generating credentials for connecting to the SSH Proxy.

Security Fixes

- Fixed XSS issue on Secret Share.

Release Notes 10.2.000018

Release Date: 5/17/2017

Enhancements

- Added additional RDP Launcher type to facilitate future customization.
- The connection bar text will show the target machine's address in a tunneled RDP session.
- Added the ability to search for a Secret Template by name using REST.
- Added the ability to set Secret permissions using REST.
- Added the ability to delete Folders using REST.
- Added the ability for Secret Server to discover COM+ dependencies.
- Added a new Secret Template and password changer for Watchguard Firewalls.
- Distributed Engine site selection drop down will now autocomplete with 50 or more Sites.
- Added option to specify whether a domain in Secret Server will be used for logging into Secret Server.

Bug Fixes

- Fixed issue where the SSH Machine scanner would not use multiple Secrets or Filters for scanning.
- Fixed issue where Discovery machine scanners set to authenticate did not throw errors when authentication failed.
- Fixed issue where PowerShell dependency information was cleared when editing the dependency after initial creation.
- Fixed issue where REST calls to update a Secret incorrectly bypassed Request Access.
- Fixed issue where scheduled Report emails did not include the full URL to view the Report in Secret Server.
- Fixed issue where PowerShell dependency arguments incorrectly referenced Secret field display names.
- Fixed issue where creating a new PowerShell Ticket System would throw an error.
- Fixed issue where the PowerShell script tester did not reference Distributed Engine if it was enabled on the Local Site.

- Fixed issue where the Site list shown when creating a new Discovery source contains an invalid entry.
- Fixed issue where users without the Share Secret Role permission were able to share Secrets.
- Fixed issue where clicking the Back button on the View Audit page of a Secret and clicking the Back button again on the Secret view page would cause a redirect loop.
- Fixed issue where converting a Secret to a new Secret Template does not recreate dependencies.
- Fixed issue where resetting the database connection could throw an error.
- Fixed issue where Discovery did not handle CredSSP and WinRM correctly on specific Organizational Units.
- Fixed issue where session launchers would occasionally fail in a Secret Server environment where a Load Balancer was present.
- Fixed issue where saving a Secret's audit to a file would throw an error.
- Fixed issue where Discovery rules occasionally created duplicate Secrets.
- Fixed issue where Reports could be sent without specifying an email address.
- Fixed issue where Engines did not display that they were offline after failing connection verification checks.
- Fixed issue where using REST to delete a user threw an error.
- Fixed issue where Secret Server could not be upgraded using Internet Explorer.
- Added support for application account impersonation via SOAP web services.

Security Fixes

- Fixed XSS issue on Discovery Network View.
- Fixed XSS issue on Dashboard.
- Fixed Frame Blocking issue on Dashboard.
- Fixed potential security issue with the Chrome Login Assist Extension.

Release Notes 10.2.000001

Release Date: 4/25/2017

Security Enhancements

- Fixed an issue in 10.2.000000 where a highly privileged Secret Server administrator could, in certain select circumstances, be inadvertently granted read access to Secret data that is protected by Secret Workflow. This issue was found during routine internal testing and review.
- Enhanced security around various ajax calls.

Release Notes 10.2.000000

Release Date: 4/12/2017

Enhancements

Session Monitoring

Remote Desktop Metadata (May require an additional license)

- New session monitoring agent records additional data from the RDP sessions including process activity, keystrokes, and more.
- The Monitoring agent adds support for recording remote sessions on servers that were not launched directly from Secret Server.
- Updated the session search UI to support cross session searching for data within the session and additional filtering options
- Updated the session playback UI to support in browser playback and activity points in the session
- Performance enhancements to session processing speed

Discovery

- Added out of box discovery to find Active Directory accounts on the domain
- Added option to discovery import rules to limit the number of Secrets to import to prevent unexpected takeover of accounts

Upgrades

- Added a Setup console to manage upgrades and core product configuration across different Delinea installations
- New Secret Server upgrade manager that gives more detailed messages and support upgrading multiple products from a single interface.

UI Updates

- The Secret Server header has been modified for more logical grouping of menus
- Moved user specific menu items under a new user icon header
- Added a new Alert Notification Center header icon with a badge to show pending alerts.
- Removed support for customer defined HTML help pages.
- Added new in app help messages to page headers

REST

- Added endpoint for Launcher lookups
- Added Session Monitoring endpoints

SAML

- Added support for SHA256 for SAML request signing
- Added support for ForceAuth to support forcing credentials when first navigating to Secret Server even if logged into the identity provider.
- Added support for signing SAML requests in a CNG Key Storage Provider

Note: The upgrade to 10.2 will migrate the saml.config to a new format to support added features.

- Running a script test from the UI now has an option to select the Site to run it on
- Added an option to select a Secret to run a test script as
- Added internal site connector for background message processing.
- Added support for 2FA for SOAP winauth web services.
- Added timeout setting for RADIUS authentication
- Updated SSH Library for heartbeat and password changing to support more ciphers.
- Added a $$CHECKNOTCONTAINS check to the SSH password changers
- Added custom port support for SSH password changing and heartbeat
- Added default mainframe password requirement

WARNING: SQL Server 2005 is no longer supported.

- A new desktop client is available. For instructions and download links please see the Desktop and Mobile App Guide

Bug Fixes

- Fixed issue where scheduled backup wasn't working for Free edition
- Fixed HSM session leaking for Safenet Luna PCI cards
- Fixed upgrade issue that could cause database errors in some cases where discovered Dependencies were not able to properly map to a Scan Template.
- Fixed issue where the default WinRM endpoint was not used by an engine if the WinRM endpoint was left blank on the Site configuration.
- Fixed unnecessary logging in Discovery
- Fixed issue with engines not upgrading after a Secret Server upgrade
- Fixed locking errors that could occur on the file system when debug logging was enabled.
- Fixed issue with scanning specific OU's with a custom PowerShell Discovery script.

Secret Server Release Notes 10.1.x

Release Notes 10.1.000023

Release Date: 2/22/2017

Enhancements

- Added additional actions to user audit for when 2-factor is changed on the user.
- Added status icon to the Heartbeat field on Secrets. Going forward new Heartbeat and Password Change errors can be viewed in a Secret's audit log for quicker diagnosis and reporting. Note that these error messages are not backfilled so only new errors will show in the log going forward.
- Added support for multiple domain controller IP addresses in the domain field of an Active Directory Secret for cases when the domain name isn't resolvable for heartbeat and password changing.
- Updated behavior in the SOAP API for disabling Check Out on Secrets that are currently checked out to match bulk operations behavior. A Secret Owner can now call SetCheckOutEnabled to turn off check out on a currently checked out Secret.
- Added a new role permission for creating application user accounts.
- The SSH Proxy now restricts the default cipher suite for incoming connections.
- SOAP API – Added new method for GetReport to get report data via the API

Bug Fixes

- Fixed issue where password changing through Distributed Engine would not run in Professional Edition
- Fixed localization issues in logs
- Fixed engine upgrade error when upgrading from the legacy agent to distributed engine.
- Fixed issue where the database field tracking when the Secret expiration field was initially set using the server time instead of UTC
- Fixed issues with the Secret Search Filter for Discovery
- Fixed issue where getting redirected to the Logged in at other Location could cause the user to be logged out at both locations.
- Fixed issue in AD sync where an error was logged in some cases if the client was accessing from behind a load balancer.
- Fixed issue where using the Folder Slider on dashboard and deleting the currently selected folder would break dashboard search.
- Fixed issue where you could set an approval group that only contained an application account user
- Fixed exception that could occur in the system log for license expiration checks.
- Fixed issue where the only privileged account options in Secret Policy were for LDAP or Active Directory Secrets.
- Fixed issue where the Configuration Edit Event Subscription didn't fire if email settings were modified
- Fixed issue where a large custom expiration data, such as 12/1/9999 on a Secret caused 500 errors on Dashboard search
- Fixed SQL Replication issues where web server nodes connected to subscribers redirected to replication page and audit insert errors could occur.

Security Fixes

- Fixed XSS issue on Discovery Scanners.
- Fixed XSS issue on Secret View for certain launcher configurations.

Release Notes 10.1.000000

Release Date: 1/18/2017

Enhancements

SSH Key Management (May require an additional license)

- Added ability to automatically generate new public / private key pairs and rotate the public key on servers.

z/OS RACF Support (Requires Premium Edition or higher)

- Added support for automatically manage IBM z/OS RACF credentials

Dual Control

- Added options to enforce dual control when viewing recorded sessions, shadowing sessions, and running reports to enforce 4 eyes principle for potentially sensitive audit information.

New Built in Reports

- Unlimited Administrator Activity: Shows actions done by users with the unlimited administrator permission when break the glass mode is enabled.
- What Secrets Changed Passwords in Last 90 Days: Shows Secrets that have had their passwords changed in the last 90 days.
- What Secrets Have Not Had Passwords Changed in Last 90 Days: Shows Secrets that have not had a password change in the last 90 days.
- What Folders Have Policies Assigned: Shows what Secret Policies are assigned to folders.
- What Secrets Have Different Policies Than Their Folders: Shows Secrets that aren't inheriting their policies from their Folder.
- What Secrets have Policies Assigned: Shows what policies are assigned to each Secret.
- User Activity Report: Added User's current locked out status to the user activity report.
- Added ability to auto enable Google Authenticator, Duo, and email two factor as part of domain synchronization.
- Dependencies on Secrets can now be grouped so they can be assigned to different Sites when a service account is used across segregated networks.
- The Delete Secrets role permission has been split into separate permissions for delete secrets and delete secrets from reports.
- When session recordings are stored to disk rather than in the database there is now an option to encrypt the videos.

- Renamed Domain Friendly Name to NetBIOS name on Active Directory administration page.
- Application API Accounts can now log in directly to both the SOAP and REST API's

REST API

See "REST API PowerShell Scripts" on page 1503.

- **Token Expiration:**

  New expiration endpoint to invalidate an issued token
- **File Upload / Download:** Upload and download files from Secrets
- **Field Update / Get:** Get or update a specific Secret field value with a single call rather than getting the full Secret object and posting an updated Secret object
- **SSH Keys:** Added options to change password and create Secret for generating new SSH keys and passphrases.

> **Note:** As of 10.0 the REST API and SOAP API tokens are not interchangeable due to added support for OAUTH. Each API requires its own authentication call and token.

Bug Fixes

- Fixed issue where emailing reports wouldn't use the selected date range.
- Fixed issue where a backslash in the dashboard search wouldn't return any results.
- Fixed issue where scheduled backups were not available in Free edition.
- Updated the Windows Password Changer to support changing the built in administrator accounts without having to specify a privileged Secret due to Microsoft Patches 3177108 and 316769.
- Fixed issue where an admin could convert the only local admin account to an Active Directory Account.
- Fixed foreign key error in Discovery when an OU is deleted that is part of a discovery import rule
- Fixed issue where testing PowerShell scripts failed when a PSObject was returned by the script.
- Fixed issue where reports did not email with the correct date range.
- Removed special characters from SSH Proxy one time credentials to prevent issues with some custom launchers where special characters break command line arguments.
- Fixed issue where SSH Command Sets were not available in Professional Edition for Discovery.
- Fixed issue where Discovery could return an error when matching found accounts against Secrets with an inactive Secret Type.
- Fixed issue where adding an Active Directory Domain with the same name as an SSH based Discovery Source would cause an error.
- Fixed issue with REST tokens that could occur in some environments when FIPS mode was enabled.
- Fixed issue where the Database verify step fails in the web installer when Maintenance Mode is enabled. Note that since this is a change to the installer it will not take effect during the upgrade to 10.1 because the upgrade is running off of the current version.
- Fixed issues with the Mac Session Launcher when running on Sierra.

- **REST API:** Fixed permission check issues in the REST API where editing a Secret with check out enabled was improperly allowed.
- **REST API:** Fixed permission check issues in the REST API where users with View access could see the AutoChangeNextPassword field

Secret Server Release Notes 10.0.x

Release Notes 10.0.000006

Release Date: 10/20/2016

Enhancements

- Added Secret Search Filter to Discovery Scanners to dynamically find a Secret to authenticate to a machine for scanning. See this KB for instructions on creating Secret Search Filters.
- Custom PowerShell password changers are now configured and defined in Remote Password Changing rather than on the Secret Template. See this KB for updated instructions on creating PowerShell Password Changers
- Added option for matching Dependencies to Secrets based on a remote machine in addition to a domain for better support of database links and other local account type Dependencies
- Scan Item Template has been renamed to Scan Template in the Scriptable Discovery Admin UI
- Added Scan Template column to the Discovery Network View results view

Bug Fixes

- Fixed issue where launchers could periodically fail in a load balanced environment because session information was only stored on the web server the session was started from.
- Fixed issue where UNIX host ranges were not removed in the Discovery Network View after they were removed from the Discovery Source.
- Fixed issue where testing PowerShell scripts that returned PowerShell objects on the Admin Scripts page could return a 500 error from the server.

Security Fixes

- Fixed issue in REST web services discovered during internal review. Only customers running 10.0.000000 are affected. See this advisory for more information

Release Notes 10.0.000000

Release Date: 10/13/2016

- Scriptable Discovery (Enterprise Plus or Advanced Scripting Add-On)
  - Administrators can create PowerShell scripts to customize Discovery for local accounts and service accounts
  - Domain specific settings for service accounts, remote connection type, and extended account information have been moved to the relevant scanner on the Discovery Source page

  - NOTE: Custom SSH, SQL, and PowerShell script Dependencies are now managed as Dependency Templates for simplification of administration and integration with custom Discovery sources. Custom scripts will no longer be directly assignable as Dependencies on Secrets.
  - See the Scriptable Discovery Overview KB article for more information and example usage
- Distributed Proxying
  - Distributed Engines can be set to proxy Secret Server sessions as an alternative to the Secret Server web server.
- Privilege Manager for Windows
  - Secret Server and Privilege Manager for Windows can be co-deployed and share authentication and management
  - Requires separate purchase of Privilege Manager for Windows (formerly Application Control Solution)
- Added Secret as an option for the Domain Synchronization credential
- Added CAPTCHA support for logins
- Added configuration setting to prevent password re-use when changing a Secret's password.
- Added support for AES-CTR with SSH password changers when running in FIPS mode.
- Added support for MFA tokens with AWS password changing
- NOTE: Secret Server 10.0.000000 requires configuring integrated pipeline mode on the Secret Server Application Pool Please see this KB for details on configuring integrated pipeline mode in IIS. If using Integrated Windows Authentication you will also need to update IIS authentication settings as detailed in this KB.
- Step Upgrade: Upgrading to 10.0.000000 requires that you first upgrade to 9.1.000001, which has changes to the upgrader to support moving to 10.0.000000.
- NOTE: As of Secret Server 10.0 REST and SOAP API tokens are not interchangeable. Each API requires it's own authentication call and token. Bug Fixes
- Fixed issue where discovery would return an error if there was a duplicate deleted user on a windows machine.
- Fixed issues where 2-factor remember me and inactivity timeout could conflict
- Fixed issue when synchronizing cross domain groups
- Fixed issue where Remote Password Changing and Heartbeat would fail on the same machine as a Distributed Engine
- Fixed issues with checking empty fields in REST API
- Fixed issue where the REST API folder search permissions were too restrictive
- Fixed impersonation error when running a SQL Script Dependency
- Fixed issue in Audits when using mapped IPv4 addresses that exceeded 40 characters.
- Fixed issue where Password Changing, Heartbeat, and Discovery did not consistently work on the same machine as a Distributed Engine
- Fixed issue where the Syslog RT field did not respect the UTC time setting.
- Fixed issue with engine licensing enforcement.

- Fixed issue where a foreign key constraint from a deleted Discovery Rule could stop Discovery
- Fixed issues with SonicWALL password changers
- Fixed incorrect text warning when creating an Application Account
- Fixed impersonation issue with SQL Dependencies
- Fixed issue where the delete action on Event Subscriptions could delete the incorrect row. Security Fixes
- Fixed Open Redirect issues on multiple pages
- Fixed XSS issues on multiple pages
- Added an upper limit to local user passwords to prevent a denial of service attack with extremely long passwords
- Fixed issue where Distributed Engine did not work when restricted to TLS 1.2
- Fixed issue with MS SQL password changing where the new password showed in SQL Trace on the target database server Are you an IBM Security Secret Server customer? Access IBM-specific documentation here.

Secret Server Release Notes 9.x

Release Notes 9.1.000001

Release Date: 10/13/2016

Enhancements

- It is required to upgrade to 9.1.000001 before Secret Server will upgrade to 10.0.000000
- Added installer enhancements to support the 10.0.000000 release. Release Notes 9.1.000000 Release Date: 7/13/2016 Enhancements
- REST API
  - REST based web services API for managing Secrets, Users, and Groups.
  - For more information see the REST API Guide on the Secret Server documents page
- Web Password Filler
  - A new Chrome extension for website logins is available, for more info see this KB article.
  - NOTE: After upgrade, Chrome users will be prompted automatically to install this extension. Firefox and Internet Explorer users will continue to use the existing add on or bookmarklet.
- Site per OU in Discovery
  - Assign an Engine Site at the OU level in Discovery
  - Set a different Secret per OU in Discovery
- Added option to set owners on user accounts to delegate account management
- Added support for SCP through the SSH proxy
- Added additional options to the Secret Expiration event subscription
- Disabled dependencies are hidden by default on the Secret Dependency page

- Added additional option for windows password changers to help handle multiple IP addresses in DNS for a single machine
- Editing a password field on a Secret with password changing enabled now gives the user a dismissable prompt to help prevent mistaken password edits
- Domain user accounts can now be marked as Application Accounts for integrated auth web service access only

Bug Fixes

- ConnectWise integration now uses the API rather than database table integration. See this KB for information on setting up API access to ConnectWise.
- Fixed issue where multiple syslog destinations using the FQDN did not work
- Fixed issue where a user viewing a Secret after a password change within the Secret View interval after their last Secret View did not result in an audit.
- Fixed issue where Oracle error ORA-12170 was treated as heartbeat failed rather than unable to connect.
- System log truncation notification email goes to users with Administer System Log permission rather than Administer Configuration
- Fixed issue where commas in group names were not parsed correctly on AD Sync
- Fixed issue with AD sync when a group had more than 1500 members
- Fixed issue with AD sync when the OU has asterisks in the name
- Fixed issue where Session launchers did not trim spaces from username and machine fields
- Fixed syslog error when the event details exceeded 4000 characters
- Performance updates for the Recents Secrets widget and Secret Load when there are a large number of audit records on a Secret
- Check In web service method now respects the Force Checkin role permission.
- Fixed access denied message when doing a bulk operation for convert secret template without the view deleted secrets role permission
- Fixed potential licensing error when running the PowerShell password changer
- Fixed issue where setting AutoChange schedule through Secret Policy would not use UTC
- Added support for HMAC-SHA2-256 and HMAC-SHA-512 ciphers for SSH Heartbeat and Password Changing
- Fixed issue with SSH dependencies on Cisco devices where the setenv command was not available
- Added additional information to the Subscription Dependency failure email to include machine name and dependency name that failed
- Added additional logging for Heartbeat and Password Change monitors Mobile Updates
- The Delinea PAM Android app has been republished. Existing Android users will need to uninstall and re-install to get the new version.

Release Notes 9.0.000000

Release Date: 4/13/2016

Enhancements

- Mac Session Launcher
  - RDP, SSH, and Custom Launchers are now supported with the new Mac OS X protocol handler.
  - For more information see this KB.
- Geo Replication
  - MS SQL Replication is now supported as an additional add on module. Contact your account rep if you are interested.
- UNIX Privilege Manager
  - Administrators can configure SSH command menus to limit what users can do with root and other privileged credentials.
  - Requires a separate add on, contact your account rep if you are interested.
- Remember Me is now available for 2 factor.
- New option for SSH launchers to specify a Connect As Secret to make the initial connection before switching to the current Secret's user for cases when accounts are denied SSH login.
- Dependencies and Secret Audit are now copied to the new Secret when converting Secrets.
- The Tree View on Dashboard and Discovery Network View is now collapsible.
- Windows Discovery now finds:
  - If an account is Local Administrator
  - If an account is in the Local Administrators Group
  - Password last set date
  - Password expiration date
  - Password expiration status

Bug Fixes

- Fixed issue where domain FQDN wasn't populated during Active Directory Sync.
- Fixed issue with syncing an Active Directory Group with more than 1,500 members.
- Fixed issue where SSH proxy wouldn't restart after web server failover.
- Fixed issue where searching wouldn't work on Secret name's starting with ":"
- Fixed issue where selecting an approval user or group could cause an error on Secret Policy creation.
- Added optional remember me setting for two factor authentication.

Security Fixes

- The version of PuTTY shipped with Secret Server has been updated to version 0.67 to include the latest security fixes. For more information please refer to the PuTTY change log.

Secret Server Release Notes 8.x

Release Notes 8.9.300008

Release Date: 3/8/2016

Enhancements

- Secret Script Dependency Parameters can now reference associated Secrets by Secret ID in addition to the Secret order number in the associated Secrets list. See this KB for more information.
- Added new Time to Live and Retry Time settings to Distributed Engine configuration
- Secret Server Express Edition is now called Secret Server Free. There are no changes in capabilities available between the two editions.

Bug Fixes

- Fixed issue where domain password changing failed when target credential was on different domain than Secret Server and no privileged account was used
- Fixed issue with running Discovery over LDAPS
- Fixed issue where nested groups would not import correctly in AD synchronization when the group is nested within multiple AD groups
- Fixed issue where Folder was not added to the Dependency when importing Scheduled Tasks through Discovery
- Fixed issue where scheduled task discovery could get incorrectly marked with an error and prevent import
- Fixed authentication issues when using the Web Password Filler with Integrated Windows Authentication
- Fixed RDP proxying error when using FIPS compliance mode
- Fixed Session Launcher error if TLS 1.0 is disabled on the web server.
- Fixed Discovery issue when scanning using credentials from a different domain.
- Fixed issue where new domain users were not getting a personal folder.
- Fixed issue where Distributed Engine could create excessive database entries for background threads
- Oracle Script Dependencies will now ignore extra parameters passed in from Secret Server
- Fixed potential error during upgrade if there were users that had never logged in

Security Fixes

- Fixed reflected XSS issue
- Removed ASP.NET version disclosure from response headers

Release Notes 8.9.300000

Release Date: 1/13/2016

Main Focus: Active Directory Synchronization Through Engine

- Active Directory sync through Distributed Engine
  - Active Directory synchronization and user authentication can now be routed through a specified site. This allows for AD authentication even if the Secret Server web server does not have direct access to the domain.
- Password Requirements now support starting character rules.
  - When target systems disallow certain characters, users can now set a rule for which characters a generated password is allowed to start with.
- Dates are now stored in UTC format
  - Customers with servers in different time zones no longer need to set the servers to use the same timezone or UTC time. Existing dates in the database will be retrofitted to UTC if the web server is not already in UTC time.
- Installer updates
  - Improved installer to pre-configure IIS and .NETfor fresh installation
  - Added configuration wizard for the initial setup of Secret Server
  - New users will see a dashboard overlay highlighting key features.
- Added configuration option to allow for concurrent login sessions.
- The session launcher .NET framework support has moved from .NET 3.5 to .NET 4.5.1 and higher.
- Added configuration option to enable frame breaking.
- FIPS support is now available in Enterprise Edition.

Bug Fixes

- Fixed issue where local windows account heartbeat and password changing didn't work on the same machine as an engine.
- Fixed issue where ticket links weren't clickable in audit logs when generated by an access request.
- SOAP web services now respect the ZeroInformationDisclosureMessage setting recommended in the Security Hardening Report.
- Fixed issue where local account discovery scanned domain controllers in some scenarios.

Security Fixes

- Fixed security issue with named pipe permissions when passing credentials to the PuTTY launcher.
- Fixed an XSS vulnerability.

Release Notes 8.9.000022

Release Date: 10/1/2015

Main Focus: Ticket System Integration and Security Fixes

- Ticket System Integration
  - Secret Server will validate whether a ticket is open in either BMC Remedy or ServiceNow as part of the require comment and approval for access workflows.
  - Enterprise Plus customers can create PowerShell scripts to create a custom workflow or integrate with other solutions.
- API Updates
  - AddGroupToActiveDirectorySynchronization: Adds a group to the Active Directory Synchronization list.
  - RunActiveDirectorySynchronization: Kicks off the Active Directory User Synchronization Process.
  - AddSecretPolicy: Adds a new Secret Policy.
  - AssignSecretPolicyForSecret: Set a Secret policy on a Secret.
  - SearchSecretPolicies: Search existing Secret Policies.
  - GetScript, AddScript, GetAllScripts: New methods for managing the PowerShell, SSH, and SQL scripts.
  - The Folder Extended Windows Authenticated Web Service methods no longer have the token parameter.
- Added NAS attributes to the RADIUS messages.
- The SonicWALL Web Admin and SonicWALL Web Local User password changers have an option to validate or bypass remote SSL certificates.
- The RDP Session Launcher now shows the end target machine name in the RDP window when RDP Proxying is used.
- Logging to the Remote Password Change log when a Secret isn't changed because it's outside its AutoChange Scheduled time is now only logged once.
- Added new option for Active Directory Discovery Sources to resolve based on machine name only.
- Added new options for how the custom process launcher runs to help handle UAC prompts.
- SSH, SQL, and PowerShell Dependencies can now use the $CURRENTPASSWORD token.
- Updated the web password filler to prioritize exact matches in the search results to help show matching Secrets when on sub-domains.
- IP Address Restrictions can now be applied to Active Directory Groups.

Bug Fixes

- Fixed issue where Dashboard would not display in Firefox 41.
- Fixed performance issue some customers were seeing after upgrades to 8.9.
- Fixed issue where SSH Dependencies were suppressing the full error details.
- Fixed issue where SSH connections were not being closed after Heartbeat.
- Fixed test dialog for custom UNIX password changers with linked Secrets.
- Fixed incorrect display of the SSH Log link in Secret Audit trails.

- Fixed issue where pressing enter in the quick search area when viewing a Secret would run the Secret Launcher in some browsers.
- Secret Server will no longer override root level IIS HTTP Redirects on upgrade.
- Fixed issue where the Web Password Filler didn't work with SAML integration.
- Fixed error in test dialog for custom UNIX password changers when no key was present.
- Fixed copy to clipboard issue in IE11.
- Fixed issue where hitting Enter on Secret Edit would prompt to generate a new password.
- Fixed password strength error alert on Secret View.
- Fixed issue where SSH Discovery would leave hanging sshd processes on AIX instances.
- Fixed issue where duplicate Active Directory discovery sources could be created. Security Fixes
- Fixed security issue with update checks with update process. See our security advisory for more details.
- NOTE: It is recommended to perform an offline upgrade to 8.9.000022. See this KB article for instructions on performing offline upgrades. Upgrading Without Outbound Access
- Fixed DOM XSS issue.

Release Notes 8.9.000000

Release Date: 8/7/2015

NOTE: Secret Server version 8.8 will be the last version to support Windows Server 2008. If you wish to upgrade to a version higher than 8.8, you will need to upgrade your server to Windows Server 2008 R2 or higher.

Main Focus: Distributed Engine

- Distributed Engine – SITES
  - Distributed Engine is a NEW feature. All existing customers will receive unlimited Sites to replace our Agent feature. A Site can be assigned to a Secret or a Discovery Source.
  - Discovery can be run through the Sites to provide discovery on remote sites.
  - Customers using Agents will need to install an additional service. Review this KB and this KB prior to upgrade. ▪PLEASE NOTE: After upgrading, Secret Server will automatically upgrade all Agents to Sites. Agents will not be available after upgrading to 8.9.
  - API Change: The web service method "AssignToAgent" has renamed to "AssignSite". Use the new method, or use Secret Policy to assign Sites to Secrets.
- Distributed Engine – ENGINES
  - All existing customers will receive enhanced performance through our new Engine technology. Engines are installed on remote networks and are grouped by Site in Secret Server. The new Engines will provide improved performance for Heartbeat, Remote Password Changing, and Discovery. See this KB for additional information.
- RDP Proxying

  - RDP Sessions can now be proxied through Secret Server.
  - Secret Proxying can now be set per Secret and in Secret Policy, as well as through the API.
- Advanced Permissions
  - Several new Permissions have been added and the folder and Secret Permission UI enhanced. Permissions on folders and what Secrets inherit can now be set separately. ▪ List Folder – Allows user to traverse a folder without seeing the contained Secrets. ▪ Add Secret – Allows a user to Add a Secret to a folder. ▪ List Secret – Allows a user to see that a Secret exists and view the audit, but not see the Secret contents
- Added support for literal arguments in SSH Dependency Scripts.
- Custom icons can now be set on custom launchers.
- Added new #FOLDERID and #FOLDERPATH parameters for custom reports. API Changes
- New API methods
  - FolderExtendedUpdate – Allows updating a folder with permissions and policy.
  - FolderExtendedGet – Retrieves an existing folder with extended settings.
  - FolderExtendedGetNew – Retrieves a new blank object.
  - FolderExtendedCreate – Add a new folder with permissions and policy settings.
  - Impersonate – Allows web services impersonation of other users for API integrations. Requires the new "Web Services Impersonate" role permission be assigned and that the target user approve the request.
- Updated API Methods
  - AddNewSecret, GetBlankSecret, GetSecret, and UpdateSecret have been updated to account for new permissions. These methods will continue to be backwards compatible, but it is recommended to review the WSDL prior to upgrading if making use of these methods

Bug Fixes

- Fixed issue with SSH Proxying when using the Safenet HSM.
- Fixed issue where the IsFile element in the XML export was not properly set.
- Fixed issue where SSH Dependencies would attempt to use a password first even when a key was set.
- Fixed issue where the Dependency Discovery Import did not apply Secret Policy for newly created Secrets.
- Fixed issues with web password filler in IE 11 enterprise mode.
- Fixed issue where testing SSH scripts would not use a test SSH key for authentication.
- Fixed memory issues in Scheduled Task Discovery.

Release Notes 8.8.000020

- Fixed an XSS vulnerability. For more information, see our Security Advisory.
- Added option for SIEM messages to use UTC date instead of Server Date.
- Added an option to load the user profile when running custom launchers.
  - If you have deployed the protocol handler through Group Policy to your users, it will need to be updated.

- Fixed issue where web password filler would not recognize some password fields correctly.
- Added new web service methods for searching Secrets by exposed fields.
- Fixed an error that would happen if an SSH key was not provided when testing custom SSH remote password changing commands.

Release Notes 8.8.000018

Release Date: 3/16/2015

Enhancements

- Added Per Secret Key Encryption
- Administrators can rotate these keys periodically (Enterprise Plus). For more information please refer to this KB article on Secret key rotation.
- Updated local user hashed passwords to use PBKDF2 going forward.
- Administrators can now choose an RSA key size when configuring the HSM integration.
- Managing Dependencies on a Secret now only requires Edit access to the Secret. Importing Service Accounts from Discovery requires Edit on the Folder the Secrets will be created in.

Bug Fixes

- Fixed issue with Daylight Savings time offset in approval for access.
- Fixed issue where the bookmarklet would return Secrets that did not have URL fields.
- Fixed issue with importing duplicate Secrets with the XML import.
- Fixed issue with Google Auth two-factor when HSM is enabled.

Release Notes 8.8.000005

Release Date: 2/20/2015

- Fixed an XSS vulnerability. For more information, see our Security Advisory.

Release Notes 8.8.000004

Release Date: 2/10/2015

- Added new extended mapping for specifying a public key digest when connecting to a server for password changing, Heartbeat, Discovery, or through a Launcher. If the public key digest is present, it will be validated. For more information, see our Security Advisory and KB article on how to add public keys.
- Fixed performance issues with Web Password Filler, caused by many Secrets containing matching URLs.
- Fixed issue where Secrets that have an Auto Change Schedule might not change if there are many Secrets failing password changing.
- Fixed issue where the Regex file dependency wouldn't work with a privileged account on an untrusted domain.
- Fixed issue where Active Directory Synchronization wouldn't find users on a domain if a group being synchronized had zero members.

Release Notes 8.8.000001

- Fixed IE 8 compatibility issue. Release Notes 8.8.000000 Main Focus: SSH Key Support and Dependency Scripting
- SSH Key Support
  - SSH Keys are now supported for authentication with PuTTY, Dependencies, Remote Password Changing, and Discovery.
  - Added a new SSH Key Secret Template and added Key and Passphrase Fields to default UNIX Secret Templates.
- SAP
  - Updated the SAP libraries used by the SAP Password Changer NOTE: In order for SAP Password Changing to work after an upgrade, the SAP libraries on the Secret Server instance need to be updated. Please follow the steps in this KB.
- Dependency Updates
  - Admins can now create SSH and SQL Scripts to run as Dependencies in addition to the existing PowerShell Dependency types
  - The Dependency UI has been reworked for information density in cases when there are lots of Dependencies for a single Secret
  - Dependencies can now be retried and additional logging is now available per Dependency
  - When updating Dependencies for an Active Directory Account Secret, Secret Server will try to automatically unlock the account if it gets locked out, if there is a privileged account set on the Secret.
- HSM
  - Thales HSM's are now supported
  - Safenet Network HSMs are now supported.
- Administrators can use custom created PowerShell scripts for password changing.
- Added a new Office365 password changer.
- File Attachments now can keep history.
- New API methods
  - SearchUsers
  - GetUser
  - UpdateUser
  - GetSecretItemHistoryByFieldName
- Added a new widget for managing access requests.
- Approvers can now set a start time for an approval for access request.
- Approvers are now required to enter a reason when approving an access request within Secret Server.

- Added a new role permission Administer Create Users for creating users only. To edit user accounts, administrators will still need the Administer Users role permission.
- Maximum Attempts can now be set for Password Changing on the Secret Template.
- A custom field for displaying to users on the Basic Home can now be set on the Secret Template.
- The Protocol Handler is now the default launcher option for fresh installations of Secret Server.
- Computers not in specified OU's for an Active Directory Discovery Source will no longer be shown on the Discovery Network View.
- Added enrollment URL for Duo authentication for when the user is not enrolled.
- Added support for control characters in the SSH command sets.
- Added support for Secret values in the Approval for Access email customization.
- Added a Administer Create Users role permission which gives user account creation permissions only. Administer Users role permission still allows an admin to create and edit user accounts.
- Added View Audit button on the Dashboard Secret view for users that have the View Audit role permission but not the View Secret role permission.
- Syslog change: Syslog events now pass the Username instead of the Display Name of the user. Display Name has been moved to cs4 and cs4label fields. Please refer to the syslog guide for full field listing.
- NOTE: 8.8 supports running Secret Server on Windows Server 2008, but support for this will be deprecated in a future version of Secret Server. Server 2008 R2 will continue to be supported.
- Fixed an issue that would allow users with permissions to view a Secret to access the password history directly without going through Check Out or Approval for Access flows THY-Secret Server-002.

Bug Fixes

- Custom proxied SSH Launchers can now use custom fields in process arguments
- Fixed issues where Secrets created through the web password filler would not respect default field values or Secret Policy settings.
- Fixed issues with folder searching in some dialogs.
- Fixed bug where an admin could not add application accounts if the user count was already at the licensing limit.
- Fixed issue where the some OU's could not be selected in a Discovery Source when there were several OU's named similarly on the domain.
- Fixed issue where a failed password change on check in would write additional audits for Secret Set for Check In.
- Fixed memory issues in scheduled task discovery.
- Updated the query to retrieve computers from the domain to only return computers in specified OU's.
- Fixed issues with Active Directory Sync connection failures potentially disabling users.
- Fixed issue with using the attempt user password setting for RADIUS and integrated windows authentication.
- Fixed issue when creating a folder shared with hundreds of users and groups.
- Fixed workflow issues in web password filler when a Secret has check out or other security settings applied.

- Fixed issue where web password filler would not work properly if the URL was extremely long.
- Tokens are now supported for use with Duo Security.
- Events written to the Windows Event Log now have unique identifiers.
- Fixed performance issues in dashboard searching for deep folder structures.
- Fixed searching behavior where a found value is on multiple Secret fields.
- Fixed issue in dashboard searching where a backslash in the search terms would not return results in Firefox only.
- Fixed display issue on Service Account Discovery when using an account to run the scan on a child domain.
- Fixed URL encoding issues on the Basic Dashboard.

Release Notes 8.7.000000

Main Focus: ESX/ESXi and Unix Account Discovery

- Unix Account Discovery
  - In addition to Windows Local Account and AD Service Account Discovery, Secret Server can now scan and import Linux local accounts.
- ESX/ESXi Local Account Discovery
  - Discovery has been expanded to support scanning and automatically importing local accounts on ESX/ESXi systems.
- ESX/ESXi Password Changing
  - Added a new ESX Secret Template and a new ESX password changer to perform changes via VMware's API. SSH is no longer required to be enabled on the ESX/ESXi system if this password changer is used.
- Search Updates
  - Multiple search terms will use implicit ANDs rather than ORs for more accurate results.
  - Reduced the number of search hashes created in the database to help limit database growth.
  - Improved performance of searching on unencrypted Secret fields.
- There is a new option to delete secrets shown in a report.
- Added password masking in all entry fields
- Folder deletes and renames are now audited.
- RADIUS authentication now handles multiple consecutive access challenges.
- Added support for Duo Security as a two-factor option.
- Added support for optionally using a user's login password as the RADIUS password if prompted.
- Added search bar for web password filler to filter returned Secrets.
- Unmasked passwords on Secrets now use a different font to help distinguish between certain similar characters.

- Added option to specify a Secret for running Discovery in Active Directory Sources rather than using the Active Directory Synchronization credentials.
- Added "Password Changed" event subscription event.

Bug Fixes

- SSH Proxy now respects the client terminal type settings.
- Users can now edit notes fields in cases where they do not have access to the privileged account on the Secret.
- Fixed an issue where the launcher may not start when configured to use a protocol handler in Chrome and Firefox.
- Users will be able to see the name of the privileged account on the Secret if they do not have access to it.
- Logging in via the Windows Authenticated Web Services now sets the Last Login on the user.
- Enable Approval from Email is no longer on the Security Hardening report for editions without Approval for Access available.
- Fixed issue where an admin in unlimited admin mode would bypass entering in a comment when both Check Out and Require Comment were enabled on a Secret.
- Fixed issues with the Web Password Filler in IE8.
- Fixed issue where failover with the web servers could occur even if clustering was disabled.
- Fixed issue where there were inconsistent permission checks for adding and deleting between the web interface and the web service methods.
- Fixed issue where the MSI installer would not detect a local SQL 2014 instance.
- Fixed issue where a file could be uploaded to a non-File field using the web service API.
- Fixed issue where service account import could fail because the saved folder no longer exists.
- Fixed issue where Check Out and Require Comment workflows could send a user back to the dashboard instead of to the Secret.
- Fixed error where email report options were available when no SMTP server was set.
- Fixed issue where the SalesForce password changer would not correctly work on sandbox instances.
- Fixed incorrect display of line breaks in Notes fields on the Basic Dashboard view.
- Windows account discovery now uses the LastLoginTimestamp AD attribute rather than LastLogin to better support replicated domains.
- Fixed performance issues on Dashboard when loading large numbers of Secrets.
- Fixed issue where Access Request approvals could not be accessed by Email.

Release Notes 8.6.000010

Main Focus: Security Update

- Fixed an issue that would prevent the Windows Remote Desktop Launcher from cleaning up generated RDP files, which contain DPAPI encrypted passwords. This report was acknowledged within 24 hours. CVE-2014-

4861.

- Fixed an issue that would prevent users in certain time zones from viewing SSH Proxy logs.

Release Notes 8.6.000009

Main Focus: Security Update

- Fixed security issues reported by a customer. This report was acknowledged within 24 hours.
- Added built-in support for HTTP Strict Transport Security (HSTS).
- Improved performance of loading dashboard for very large installations.
- Administrators can now disable HTTP GET functionality for web services.
- Added additional HTTP headers to improve Secret Server's security policies.
- Added additional options to the new Theme Roller to change font size and padding between elements.
- Added new web service methods for adding dependencies to Secrets.

Bug Fixes

- Fixed issue where users with non-ASCII characters in their username could not be issued a valid token for web services.
- Fixed issue where Discovery scanning may not occur at expected times due to Application Pool recycles.
- Fixed issue where Windows Authentication web services did not respect the Require Two Factor for Web Services configuration option.
- Fixed issue where the Agent installer would incorrectly report the .NET Framework was not installed when the .NET Framework 4.5.2 was installed.

Release Notes 8.6.000000

Main Focus: UI Refresh and Secret Policy

- Secret Policy: Administrators can now define a policy for Secret Security and Auto Change settings. This can be applied at the Folder level and Secrets in that Folder automatically inherit those settings.
- The Secret Server UI has been significantly updated for look and feel, including a new basic dashboard view for non-admin users who just need core functionality.
  - Added a theme roller for creating new themes and uploading corporate logos.
  - Warning: Users with custom themes will be moved to the default theme on upgrade and will need to use the new Theme Roller to create a theme.
- Added Personal Folders option for users to store work related Secrets. These are only accessible by a named user by default, but can be accessed in Unlimited Admin mode by an administrator.
- Added support for mobile app authenticator soft tokens for Two-Factor.
- Added a built in SSH password changer for F5 root accounts.
- Added a Salesforce password changer. See this KB article for more information.

- DoubleLocked Secrets can now be accessed through web services.
- Added a new option to run Local Account Discovery using WMI, which can provide a performance boost in some environments where WMI is properly configured.
- Added optional Domain Controller field to the LDAP based Password Changers: LDAP (Active Directory), LDAP (openLDAP), and LDAP (DSEE).
- Reorganized the bulk operation drop down list for usability.
- Added AssignUserToGroup and GetAllGroups API methods.
- When proxying is enabled users can manually make a connection to Secret Server using the get proxy credentials API method or button on Secret.
- SSH Proxying can now be specified on a per node basis for clustered environments.
- Check Out and Approval for Access end times are now synchronized. A user will not be able to keep a Secret checked out past the approval period end time.
- Added in a configuration option for whether launched sessions automatically close on Check In.
- Added additional logging and event subscriptions for when DPAPI encryption is enabled or disabled.
- Improved performance for the SearchSecrets API call.
- Cluster computer objects are now ignored by default in Discovery.

Bug Fixes

- Added extra error handling to the Discovery process.
- Fixed issue with running user audit report with the Exclude Changed and Deleted Secrets.
- Updated the web password filler to handle different zones in IE. Due to security restrictions users may now be required to log in to the web password filler in addition to Secret Server. Other browsers are unaffected.
- Fixed performance issues in reports with large amounts of data.
- Fixed issue where the Secret Export incorrectly reflected the Secret count for a Folder.
- Fixed date range search in Session Monitoring.
- Fixed issue where automatic backups were not available in Express Edition.
- Fixed issue with email two-factor in Express Edition.
- Fixed issue where an incorrect SMTP configuration could cause an Application Pool Recycle.
- Fixed issue where bat file launcher would require a port field when mapping to the Secret Template.
- Fixed issue where bat file launcher did not handle parameters enclosed in double quotes correctly.
- Added performance enhancements for session video processing.
- Secret fields marked as Exposed for Display on the Template will no longer have their history encrypted for consistency and reporting.
- Fixed paging on Report Schedule History grid.
- The Out of Sync Report now shows the reason in the saved report.

- Added additional error handling for RADIUS authentication.
- Added additional error handling for Discovery machine scanning.

Release Notes 8.5.000000

Main Focus: Session Monitoring and SSH Proxying

- Upgrade to .NET Framework 4.5.1: This will require downtime and a manual change of the application pool. .Net 4.5.1 is a prerequisite for the web server. You will need to make other changes, see Considerations for Upgrading to 8.5 for details.
  - .NET 4.5.1: Secret Server now runs on .NET 4.5.1 to provide better support for the latest Microsoft technologies. To find out what this change means for you, view our KB Article.
  - PowerShell 3.0: Changes were made to the PowerShell scripting in order to fix certain remote authentication issues. These changes require an update to PowerShell 3.0.
  - Agent: If using the Agent, .NET 4.5.1 will need to be installed on machines where the Agent is installed.
  - Step Upgrade: Before upgrading to the 8.5 release, you must be running 8.4.000004. The Secret Server updater will update you to 8.4.000004 first, then allow you to update to 8.5
- Session Monitoring: The Session Monitoring administrators can now view sessions launched from Secret Server, watch activity, and even terminate the session or send a message to the end-user while the session is in progress.
- SSH Proxy: SSH Launchers can now be proxied through Secret Server. Admins can review full SSH logs of proxied sessions as part of the Session Recording feature.
- Discovery and Password Change Performance: Speed of Discovery scanning, password changing and Heartbeat checks are significantly faster for management of very large environments.
- Session Recording Retention: New configuration options are available for moving stored session movies out of the database and establishing a retention period.
- Group Owners: Owners can now be assigned to local groups. Group owners can manage membership for the group.
- Added support for PostgreSQL password changing.
- Added support for custom ODBC based password changing.
- Session Recording now uses differential images to reduce network bandwidth and database size.
- Added new Video Codec option for Microsoft Video 9, which provides high levels of compression.
- Secret Audits now include field and setting names that were changed.
- Automatic Backups now support Copy-Only database backups.
- User Audit report now has option to exclude deleted Secrets.
- Added new search options to help performance for choosing groups for Active Directory Synchronization.
- User drop down on User Audit report will properly switch to an autocomplete based on user count.
- Passwords are now masked on Secret Edit.

- Secret Check In will now terminate any open launched sessions.
- Added configuration option to check in Secrets when a launcher session is closed.
- Added P3P policy to help with cross domain issues with the Web Password Filler in IE.
- Added new configuration option to specify a custom Secret Server URL for use by the Session Launchers and Emails. This is for cases when Secret Server is behind a proxy or load balancer and a client machine cannot resolve the Secret Server web server name.

Bug Fixes

- Fixed issue with Scheduled Task Discovery on Windows Server 2003.
- Added additional checks to installer to help validate access to update files.
- Fixed a performance issue with Service Account Discovery attempting to resolve domains.
- Fixed issue with searching inside Folders on Dashboard with query string parameters.
- Fixed improper display of Edit button on custom reports.
- Web service view audits now respect the Secret View interval in configuration.
- Fixed issue where disabling check out did not clear the user it was checked out to.
- Fixed issue with bulk operation for Set Privileged Account when setting to "Credentials on Secret".
- Fixed issue where user could get an error on the Hooks tab of Check Out Secrets when not assigned the Owner permission.
- Fixed issue in 8.4 where scheduled task dependencies could be disabled from Service Account Discovery. If the instance has Service Account Discovery for tasks running these dependencies will be re-enabled. Please contact support if there are issues with Scheduled Task dependencies staying disabled.

Release Notes 8.4.000004

Main Focus: Usability and Configuration Enhancements

- Administrators can now require ticket numbers or comment for Secrets with Require Comment and Approval for Access enabled.
- The Require Comment interval when viewing a Secret can now be set on configuration so users are not prompted multiple times when accessing a Secret for the same reason.
- Added configuration option to require two-factor for API and Web Access separately.
- Added new whoami web service method to the standard web services to return what user a token is for.

Bug Fixes

- Fixed variable replacement for custom launchers in some cases when field names contained other field names.
- Added additional database connection properties for MS SQL Always On configuration.
- Fixed issue where the background processing of expired Secrets for password changing could overwrite changes in the UI in certain cases.

- Fixed issue where a custom report with a Secret ID column would cause an error if there was a row with no Secret ID value.
- Added performance enhancements for the GetSecretsByFieldValue web service method.
- Fixed potential upgrade issue for customers upgrading from versions below 7.9.000012.
- Fixed issue where copy to clipboard for Internet Explorer 10 and 11 would cause the page to scroll to the top.

Release Notes 8.4.000000

Main Focus: Service Account Discovery and Launcher Enhancements

- Multiple Launchers
  - Secrets can now have more than one Launcher, so if the same credential is used to run different tools admins can set up multiple Launchers per Secret Template.
- Added support for scanning for Scheduled Tasks and IIS Application pools as part of Service Account Discovery.
- Auto-Create Dependencies (Enterprise Plus)
  - Secret Server can now automatically link any found IIS Application Pools, Windows Services, and Scheduled tasks as Dependencies to existing Secrets.
- User added Dependencies that don't exist on the machine are now shown on the Discovery grid.
- Added new Bulk Operations
  - Heartbeat Run Now
  - Heartbeat Enable / Disable
- The Secret Server Launcher can now be optionally run using a Protocol Handler instead of Microsoft ClickOnce. This may be needed in some virtualized environments where ClickOnce does not function properly. You can read about the Protocol Handler configuration here
- Added performance improvements for Dashboard search.
- Added option to force expire Secrets from any report with a Secret Id column.
- User Bulk Operations are now available.
- Added new User preference and Secret preference for the size of the launched Remote Desktop Window.
- Web Service Change: The Secret object used in the Web Service API has new fields in the SecretSettings section for setting privileged Secrets for RPC. This is documented in the Web Service API Guide .
- .NET 3.5 SP1 Support
  - This will be the last minor version of Secret Server to run on .NET 3.5.1. The next subsequent minor version (8.5) will require the .NET Framework 4.5.1. You can read more about why this move is happening in this KB Article

Bug Fixes

- Secret IDs on reports are now links, not link buttons.
- Reports on Dashboard now show rows with background colors if specified.

- Fixed error when viewing a secret set for check out by the bulk operation and a next password was already specified.
- Fixed issue where viewing the password history would not produce an audit for password displayed.
- Fixed issues with password changing for Oracle accounts without the Alter User privilege.
- Fixed potential issues with Service Account Discovery importing duplicate dependencies.
- Fixed issue where the password strength indicator on Secret View could be incorrect.
- Fixed issues with Dependencies not matching correctly in Discovery if the username format was different.
- Fixed issues with Service Account Discovery import not properly matching to existing Secrets.
- Fixed issues with Local Account Discovery rules importing accounts from OUs excluded from the domain level scanning.
- Individual computer discovery scan logs are now limited to the number of entries stored to prevent excessive database growth.
- Fixed issue where the search results on Dashboard could sometimes be incorrect due to timing of search.
- Fixed issues with the header search box ignoring custom columns in the returned results.
- Fixed issue with an incorrect validation for Folder permissions when saving a Secret through web services.
- Fixed issue where the password strength icon on Secret View was incorrect in some cases.
- Added missing Check In method to the windows authenticated web service API.
- Fixed issue where the Check Out information was not correctly populated by the return value of the GetCheckOutStatus web service method.
- Fixed issue with enter key not starting the launcher when a drop down list was used for the target machines.

Release Notes 8.3.000019

Main Focus: SAML Support

- Added support for SAML 2.0 for authentication to Secret Server. Additional information on configuring SAML can be found here.
- Added configuration option to allow approval or denial of access requests directly from the email notifications.
- Updated Discovery to use the DNS name of the target machines for environments where that differs from the machine name.
- Added an additional configuration option to allow a separate timeout option for API sessions.
- Added the option to set a custom password requirement on the Secret.

Bug Fixes

- Fixed several places that had double encoded HTML.
- Fixed issue with the Create button getting disabled in some cases when making a new Discovery Rule.
- Fixed searching issue with Discovery Rules when searching in Child OUs.
- Fixed error exporting Secrets to CSV for large numbers of Secrets.

Release Notes 8.3.000002

Main Focus: Security Fix

- Fixed issue where administrators could export Secrets they had access to via inactive groups. This was reported by a customer and a fix was released within 24 hours.
- Exported Secret history can be viewed through this report.

Release Notes 8.3.000001

Main Focus: Bug Fixes

- Fixed issue with editing Security properties on a Secret where the Template did not have a Remote Password Changer mapped.
- Reduced timeout on Web Password Filler to streamline automatic logins where only one Secret matched.
- Added performance index for stored session images.

Release Notes 8.3.000000

Main Focus: Website Password Changing and Bug Fixes

- Website Password Changing. Secret Server now supports password changing on Amazon and Google Accounts in addition to improvements to Windows Live password changing.
- Administrators can limit Discovery to only search certain OUs for Windows Local Accounts and Service Accounts.
- Added new SonicWALL password changers for latest SonicWALL firmware versions.
- Added French Language Support
- The recipient email address is now displayed when testing email on SMTP Configuration.
- Added SearchSecretsLegacy Web Service API method to allow calls for Search Secrets via GET requests.

Bug Fixes

- Fixed issues with Windows Live password changing due to changes on Microsoft's site.
- Fixed issue where the File Dependency could get a logon failure due to privileged account username format.
- Fixed issue where Web Service authentication failed if the user did not have the View Deleted Secrets permission in some cases.
- Fixed double encoding of text in a few places in the UI.
- Save to File on the Admin Performance page now exports Fastest Time.
- User IP Address Restrictions redirects properly if navigated to with an incorrect query string.
- Fixed issue where the Discovery Import could break if an Active Directory Secret was Double Locked.
- Fixed issue when searching using Unicode characters in search terms on Dashboard.
- Fixed display issue with editing multiple file attachments on a Secret.

- Removed obsolete warning on Secret Template regarding write access to file system.
- Fixed display issues with Copy Secret button.
- Fixed issue where a required Secret File Field could be saved without an attachment.
- Added required field indicators on the Password Requirements page.
- Fixed issue where emails could be configured in Discovery Rules even when an SMTP server was not configured.
- Added validation to prevent users from enabling email two-factor when an SMTP server was not configured.
- Fixed issues with Sharing Secrets with large numbers of individual users.
- Fixed error when setting up ConnectWise integration in a new Secret Server installation.
- Added timeout to the RADIUS login page.
- Added validation for day of month when creating a Secret AutoChange Schedule.
- Fixed visibility issue with the Add Secret button on the Web Password Filler.
- Fixed issue with clear search button in IE 10.
- Fixed issue with updating Secrets via web services if some fields were left blank.
- Fixed issue with the Reset Password test action on Remote Password Changers using privileged accounts.
- Fixed performance issue in some environments when authenticating via web services.

Release Notes 8.2.000001

Main Focus: Web Password Filler Updates and Bug Fixes

- Notes Fields can now be marked as "Exposed for Display".
- The Web Password Filler will now try to automatically fill out login information even if the Secret has not been configured by an owner.
- For Heartbeat on Windows Accounts, the error condition of "RPC Service Is Unavailable" is now considered to be an Unable to Connect result.
- Webservice Functionality Change: GetSecretsByField now only returns Secret Items that have been marked as "Exposed for Display" and no longer writes an audit record for each Secret returned.

Bug Fixes

- Fixed occasional error with processing Session Recordings for certain resolutions.
- Fixed default sort order on Dashboard.
- Fixed issues with Web Password Filler in IE8.
- Fixed issue where users were not prompted to enter a comment, or request access when logging into a website with the Web Password Filler.

Release Notes 8.2.000000

Main Focus: Custom Columns

- Secret Server now requires the database to be set to 2005 Compatibility Mode or higher. Please refer to this KB article for steps on how to set that property.
- Added ability to specify custom columns on the Dashboard search. They can be Secret status information such as Heartbeat Status, or Days until Expiration, and allowed Secret Values.
- Updated and added new methods to the Web Services API. For full descriptions of the Web Services methods, please refer to the Web Service Guide.
  - SearchSecretsByFieldValue
  - AddNewSecret
  - GetNewSecret
  - UpdateSecretPermission
  - UpdateSecretPermission
  - CheckInByKey
  - Potential Breaking Change: The CheckOutEnabled property moved from Secret to the new Secret Settings section.
  - Potential Breaking Change: The GetSecret, SearchSecrets, and SearchSecretsByFolder methods now have additional parameters.
- New Audits and Event Subscriptions for Displaying Passwords, and Copying to Clipboard.
- RADIUS Two Factor can be set to be automatically enabled on new users per Domain.
- Discovery Network View now remembers the last selected tab.
- Increased performance on the Discovery Network View.
- Increased performance for Reports.
- Added optional retry interval on Secret Template for failed password changes.
- Added TimeZone configuration option.
- Added a timeout setting for automated backups.
- Inactive Users can now be selected in Reports.

Bug Fixes

- Updated the session recording video processing to work on Server 2012 x64 environments.
- Fixed issues with the XML Import / Export not applying permissions correctly when inheritance should be used.
- Fixed button layout for some resolutions on the User Edit page.
- Fixed bug where GetSecretAudit API method required Secret View permission.
- Fixed layout of Weekly and Monthly schedules for reports in Internet Explorer.
- Users can no longer click the RADIUS login button multiple times.

- Fixed paging on Discovery Network View.
- Fixed searching in Service Account Discovery log.
- Fixed potential incorrect Secret matches for Local Account Discovery when machine names were too similar.
- Discovery for Service Accounts now correctly handles the stored record if the Windows Service no longer exists or is running under a different account.
- Fixed issue where Service Account Discovery would not run automatically in Enterprise Edition.
- Fixed Windows Service Dependencies for connecting by IP Address for Local Accounts.
- Fixed bug where RADIUS could be disabled if login security settings were modified and the user didn't have permissions to the RADIUS configuration.
- The Regular Expression in the Flat File Dependency type is no longer case sensitive.
- Fixed potential exception during audit when adding large numbers of users to a group.

Release Notes 8.1.000014

Main Focus: Default Privileged Account

- Added ability to set a default Privileged Account for Windows and Active Directory Secret Templates.

Bug Fixes

- Fixed issue where personal Secret settings required Edit permission.
- Fixed bug with Copy Secret not showing field values.

Release Notes 8.1.000011

Main Focus: Web Service API & Secret Field Security

- Added Assign Agent method to Web Service API.
- Added Create User method to Web Service API.
- Added Get Secrets in Folder method to Web Service API.
- Added the ability to restrict edit access at the Secret Template Field level.
- Added the ability to set Secret Fields to not display in View mode.
- Added the ability to restrict Session Launcher computers to a specified list for when the computer is selected by the user.
- Minor display fixes on the Dashboard.
- Improved usability of the Web Password Filler.
- Sorted Bulk operations on Dashboard.
- Added the ability to set a default domain for the login screen.
- Added an 'Inherit' option to Discovery Rules to allow optional overriding of the configuration setting for created Secret permissions.

- Customers with Event Subscriptions for Configuration Edit will receive an email during the upgrade, for more information refer to this KB article.

Bug Fixes

- HSM Encryption integration fixes
  - Fixed session-use issue.
  - Fixed threading issue.
- Fixed an issue where certain event subscriptions did not fire for web services and bulk operations.
- Fixed an issue with email two factor login.
- Prevented AutoChange Schedule drift on start times.
- Improved the performance of Service Account Discovery and fixed issue due to duplicate names.
- Fixed a display issue on the AD sync user preview.
- Added an audit for Enable and Disable Role.
- Fixed issue with auto linking on the first column in Custom Reports.
- Enhanced Folder security related to root folders when being moved.
- Prevented issue where manual failover to a different web server may not occur in certain configurations.
- Fixed an issue where the Web Password Filler displayed duplicate Secrets.
- Fixed Sybase reference errors that could occur during Sybase password changing.

Release Notes 8.1.000000

Main Focus: SAP Platform Support and Languages

- SAP Platform support (Enterprise Plus)
  - A new SAP Secret Template was added to include all the fields required by the SAP Password Changer.
- Web Password Filler
  - Users can now install a bookmarklet that will fill in website login forms with Secret data. This is simpler to configure, and will work on more websites than the existing Web Launcher feature.
- Check Out Hooks using PowerShell
  - Custom PowerShell Scripts can be run as "before" and "after" actions for CheckOut enabled Secrets.
- New Languages
  - Dutch (Thank you to our partner Jan Dijk and his team at MCCS in the Netherlands for providing this translation)
  - Chinese (Simplified)
  - Spanish
  - Portuguese

- Added new API method GetSecretsByFieldValue that will return Secrets based on an exact match of a search term on a specific field.
- Increased Session Recording efficiency, movies now take up less storage in the database.
- Users can now add Folders and Edit Folders from the Dashboard.
- Users now have access to community and support resources from the Help Menu.

Bug Fixes

- Fixed bug where importing multiple service accounts created multiple Secrets.
- Fixed bug where certain special characters in the Dashboard Search could not be used.
- Fixed error where a Custom Launcher could throw an error if no parameters were set.
- Fixed bug where Admins could not disable a user with the same username but for a different domain.
- Fixed issues with PowerShell scripts impersonating as Privileged Accounts. PowerShell scripts now require that the WinRM service is configured.
- Updated the collation check on installation and upgrades to better handle different SQL language collations.
- Fixed bug where movies longer than 24 hours could not be processed.

Release Notes 8.0.000005

Main Focus: Bug Fixes

- Fixed bug where Associated Secrets for certain SSH Password Changers were hidden in the UI after upgrading.
- Fixed bug where Active Directory Groups with a symbol in the name weren't able to be synchronized.
- Fixed issues found during internal security review.

Release Notes 8.0.000004

Main Focus: Minor Improvements and Bug Fixes

- Improved long term SQL performance in heavy load scenarios.
- Fixed an issue related to privileged account visibility on the Secret Remote Password Changing page.
- Loosened collation restrictions.
- Updated contact information.

Release Notes 8.0.000000

Main Focus: New Dependencies And HSM Integration

- PowerShell Dependencies (Enterprise Plus)
  - Administrators can upload custom PowerShell scripts which can be set as Dependencies on Secrets.
  - After a password change Secret Server can execute Administrator created scripts as custom actions.
- IIS Application Pool Recycle

  - Adds the ability for Secret Server to recycle an application pool without updating the Application Pool's service account.
- New installations have an option to specify a SafeNet HSM for encryption. (Enterprise Plus)
- Added functionality for an Administrator to upload a batch file for use with a Custom Launcher.

Bug Fixes

- Fixed issue where the Launcher failed in IE in certain security zones.
- Fixed error that could appear in the system log due to OU's being deleted after the Discovery Process ran.
- Fixed duplicate checking in the CSV import.
- Fixed layout issue with the Report Widget in lower resolutions.
- Inactive Application Accounts are now hidden by default on the User Administration page.
- Fixed potential XSS vulnerability on the Dashboard.
- Fixed issues with Custom Launchers running as Privileged accounts of different Secret Types.
- Exporting reports or logs to CSV will now include the timestamp instead of just the date.

Secret Server Release Notes 7.x

Release Notes 7.9.000004

Main Focus: Security Update

- Fixed issue with launchers and Secret Check Out.
  - (This was reported by a customer – the issue was confirmed, fixed and released within 24 hours by the Secret Server team./>

Release Notes 7.9.000003

Bug Fixes

- Fixed issue that prevents upgrades on a non-default collation on the SQL Server database.
- Fixed issue where a scheduled report email would show an image link when no image was specified on the report.

Release Notes 7.9.000001

Main Focus: Layout and Bug Fixes

- Fixed display issue in Folder Tree for Bulk Move to Folder for Chrome.
- Fixed layout issues in Admin Network View for IE 7.
- The Windows Auth Web Services will now resolve an authenticated user by friendly domain name in addition to the previous authentication methods.
- Fixed error when manually emailing a report with parameters.

Release Notes 7.9.000000

Main Focus: Automatic Import of Local Accounts

- Secret Server Discovery now includes automatically creating Secrets when Local Accounts are found using "rules" (Enterprise Plus Edition)
  - Administrators can specify users that should be alerted when Local Accounts are discovered.
  - Administrators can create search rules to create Secrets when Local Accounts are discovered.
- Service Account Discovery for all Service Accounts (Enterprise Edition)
  - Secret Server will scan machines on the domain and retrieve Windows Services that run under a domain service Account.
  - Administrators can manually import these as Secrets with Dependencies, or if the Secret already exists, import the Windows Service as a Dependency.
- Linked Accounts for Custom Launchers
  - If a Secret Template is tied to a custom launcher, the owner can link other Secrets to either run the custom process, or to use for command line parameters.
- Added bulk operations for "Hide Launcher Password".
- When Unlimited Administrator is turned on, a banner is displayed on the dashboard warning users that it is on.
- Added Check In / Check Out events to Event Subscriptions and SIEM events.
- Updated error display icons to be more prominent on Event Subscription, and Password Rule screens.
- The search grid on Dashboard now expands to full screen if no widgets are in the rightmost column.
- Added installer check to prevent installation on non-compatible SQL Server collations.
- Improved performance for reports that checked Folders and Permissions.

Bug Fixes

- Fixed issue where certain unpatched versions of IE8 would not display Dashboard correctly.
- Fixed bug where th" pas"word compliance status of a Secret was not updated after a remote password change.
- Fixed issue on the Discovery page where Accounts linked to deleted Secrets were not returned when searching for Unmanaged accounts.
- Fixed error in the system log due to incorrect parsing of Dates in certain locales.
- Fixed bug where Application Accounts could be set as Secret Approvers.
- Fixed bug where Secret Owners could change Share permissions on Secrets that were set for Approval for Access without getting approved.

Release Notes 7.8.000062

Main Focus: Security/Bug Fixed

- Fixed security issue found during internal security review. (All customers are recommended to upgrade)
- Fixed locale issue on web browsers for unusual locales.

Release Notes 7.8.000061

Main Focus: Scheduled Reports

- Added scheduled reports
  - Administrators can now set up Report generation on specific schedules.
  - Reports can be emailed to a subscription list.
  - Reports can be set as "Health Checks" that will only be delivered if the conditions of the Report are met.
- Added #STARTWEEK and #ENDWEEK as dynamic Report parameters.
- Updated Active Directory Synchronization to make adding synchronization Groups in large Domains easier.
- Added Event Subscription for support license expirations. Admins can now be notified when support licenses need to be renewed.
- Updated calendar and search controls throughout the application for formatting and consistency.
- Improved inactivity timeout
  - If a tab is closed but not the browser, inactivity timeout will now work.
  - If multiple tabs are open for Secret Server, being active on any tab will prevent inactivity timeout from occurring (Except for IE).
  - If inactivity timeout occurs, all open Secret Server tabs will be redirected to the logout page (Except for IE). Bug Fixes
- Recorded IP Address in the Secret Audit record when a Dependency is updated.
- Added guard to prevent the expiration of Secrets through web services when Expiration is disabled on the Secret Template.
- Fixed the installer so it properly detects a local instance of Microsoft SQL Server 2012.
- Fixed Windows Live Password Changer due to updates on the Windows Live site.
- Updated Chrome Copy To Clipboard extension, it now installs from the Chrome web store to comply with the latest release of Chrome.
- Fixed bug where updating personal notifications for a single Secret could update personal notifications for other Secrets.

Release Notes 7.8.000048

Main Focus: Windows Live password changer and COM+ dependencies

- Added support for changing Windows Live web passwords.
- Added support for COM+ Applications as Dependencies.
- Added new Bulk Operations

  - Disable AutoChange
  - Disable Comment On View
  - Undelete
- Added Folder Name on Secret Audit header.
- Added Configuration option to prevent duplicate Secret names.
- Added name of Template created to Create Template Event Subscription emails.
- Added additional web service methods to the windows authenticated web service.
- Added Copy Secret Template.
- Added new Folder Slider on Dashboard to make navigating highly nested Folder trees simpler.
- Added additional tooltips to the Secret View page.

Bug Fixes

- Fixed issue where Agent connections could sometimes fail due to the version not being handled properly.
- Fixed issue where SQL Password Changing could fail when the target SQL instance was configured to use a dynamic port.
- Added missing audit"record for when a Secret moves to the root folder due to the Folder getting deleted.
- Fixed missing localizations on the IP Address page.
- Fixed issue where users could import Secrets without Folders when the configuration option to require Folders was turned on.
- Fixed bug where Template Name could be set to blank.
- Fixed bug where Secret permissions could get in an inconsistent state when Bulk Changing permissions and inheritance was enabled. Java API Release Notes
- Added file attachment support.

Release Notes 7.8.000040

Bug Fixes

- Added support for Next TokenCode mode for RADIUS servers.
- Fixed performance issues in Folders for IE on dashboard.
- Fixed issue where the custom commands for UNIX Remote Password Changers would not correctly parse Fields with adjacent special characters in the test dialogs.
- Fixed issue where a Secret Field specified in the Parameters value of a Custom Launcher would not get masked if Hide Launcher Password was enabled.
- Fixed incorrect display width of Folders in Folder Administration.
- Fixed duplicate Folder name shown in Reports for highly nested Folders.
- Fixed bug where OK button would not enable on folder picker for bulk operations sometimes in certain browsers.

Release Notes 7.8.000039

Main Focus: SonicWALL Integration and SSH

Enhancements

- Added support for changing passwords on SonicWALL NSA devices.
- Added support for SSH password changing where no user authentication is required to establish a connection. Used for BlueCoat Packet Shaper devices.
- CSV Import with Folder now creates the Folders if they do not exist.
- Added a column to show whether a Group is Active on the Group Membership report.
- Updated the Get Secret Audit API method to not check out a Secret if Check Out is enabled.
- Made it more clear when a folder is selected for non-default themes. Bug Fixes
- Fixed potential issue with heartbeat on SSH Secrets that would cause heartbeat to stay in pending and shut down the web application due to incompatible SSH versions.
- Fixed issue where Folders might not return in a sorted order on Dashboard.
- Fixed display issues on Dashboard for IE 9.
- Fixed bug where Configuration Change event subscriptions did not fire.
- Fixed line ending issue that caused password changing on HP iLO devices to not work.
- Fixed bug that caused Windows Authentication Web Services to not work.

Release Notes 7.8.000036

Main Focus : Application API and Ticket System Integration

- Added Application User type for use with the Application API.
- Added support for Authenticated SMTP.
- Added LDAPS support for Active Directory.
- New Bulk Operations
  - Change Check Out Status.
  - Convert Secret Template.
- New Web Service API methods
  - Secret Status to show whether a Secret is checked out.
  - Import XML to automate the advanced import.
  - Enable Check Out.
  - Expire Now.
  - Get Secret Audit.
- Discovery

  - Added new Reports for Discovery diagnostics.
  - The Full Scan log is now stored per computer.
  - Added Re-Scan button for each computer.
- Ticket System Integration
  - Administrators can enter a support system URL to navigate to Tickets from the Secret Audit.
  - Users can enter a ticket number for Require Comment and Approval for Access.
- Configuration option to change Default Secret permissions to Secret Creator only.
- Added option to allow Editors to bypass Approval for Access.
- Increased the maximum length on all Secret fields from 1991 characters to 10000 characters.
- Added new role permission for the Advanced Import.
- Increased security in the PuTTY launcher to prevent password exposure in the command line arguments.
- Added option to exclude Secrets from the User Audit Report that have been changed since the User last viewed them.

Bug Fixes

- Fixed issue when removing more than one field during a Template Convert.
- Fixed issue with Event Subscriptions Dependency Failure Events that caused the alerts to be sent every time a dependency was changed.
- Fixed issue where Application Pool Dependencies would sometimes not verify due to casing in Dependency Name.
- Added support for UTF-8 characters for the service account's password for Active Directory Synchronization.
- Added support for UTF-8 characters for RADIUS two factor.
- Fixed issue where password requirements would validate on non-required password fields.
- Updated the Automatic Backup so it will not try to delete backup types that are not enabled.
- Fixed issues with data grid paging on the Event Subscriptions screen.
- Fixed error when saving the Backup Log to a file.
- Fixed issue with Telnet Password Changer not always respecting the correct line endings.
- Fixed issue where Active Directory Group renames would not correctly resolve when synchronizing a low number of Groups.
- Fixed error on Event Subscription page when running Secret Server in FIPS compliant mode.
- Fixed display issues on Dashboard for Internet Explorer 9.
- Fixed error when returning a large number of Secrets in a Dashboard search.
- Improved email address validation for Activation.
- Improved performance on Discovery Network View.

- Fixed issue where Secrets with a 1 Day Expiration interval could change every 2 days.
- Prevented potential XSS attack on the Discovery dialog.

Release Notes 7.8.000015

Main Focus : Bug Fixes

- Fixed issue with Active Directory Synchronization for some cases where if a group was disabled, it did not get re-enabled after being resynchronized.
- Fixed issue with Active Directory Synchronization where groups with a custom schema would not be synchronized correctly.
- Fixed issue with Active Directory Synchronization where distribution groups would incorrectly get synchronized if manually added to the synchronization group list. Distribution groups will no longer work in AD sync – you must use Security Groups in AD.
- Fixed issue on Password Requirement Edit screen where a Password Requirement would fail validation if a description was not entered.
- Fixed issue with the advanced XML import where Secret data would not be created properly if there was a case sensitivity difference in the Secret Field Name and the Secret Template Field Name.
- Fixed issue with the advanced XML import where a Folder with trailing spaces in the Folder Name could be created, but no Secrets in the import would be added to the Folder.

Release Notes 7.8.000014

Main Focus : Bug Fixes and Usability

Enhancements

- Added extra detail to the Export and Unlimited Administrator email alerts.
- Added arrow key support for the Folder search on Dashboard and the quick search in the header.
- Dependency Searcher now alpha sorts machines and shows the target OS when possible.
- Added Check All option for Windows Services found by the Dependency Searcher.
- Domain and Username are remembered on the Dependency Searcher.
- Added support for updating Windows Services Dependencies that are on the same machine as an Agent or the Secret Server application.
- Added help text for IP Address ranges.
- Added explanation on the Secret Audit page and the Secret Security tab for how often View Audits are recorded.
- Added option to separately backup the application and database.
- Changed "Indexable" to "Searchable" in the Secret Template Designer.
- Added IP Address auditing for the imports.

- Modified privileges required to change a Secret's Folder. Secret Owners can change a folder regardless of whether they have the "Share Secret" permission and the Folder is inheriting permission. See the User Guidefor the full details on Folder and Secret inheritance rules.
- Removed option to specify minutes for offline access in Configuration.
- Improved error notification for the Advanced Import dialog.
- SecretID Columns are now clickable links in the Reports.
- Added Audit record for when Hide Launcher Password is changed.
- Added additional validation for Active Directory Domains to automatically resolve the Domain Name to the Fully Qualified Domain Name.

Bug Fixes

- Fixed issue with Dollar signs in custom UNIX\Cisco accounts.
- Fixed bug with large result sets when searching for linked accounts.
- Fixed issue with inactivity timeout on the server prompt for launcher for AD Secrets.
- Fixed bug where `$$CHECKFOR` and `$$CHECKINFO` commands did not work on the Password Changer test dialogs.
- Fixed issue where the Keep Alive monitor would log an error if the site certificate wasn't trusted.
- Fixed a bug where the database backups would not get deleted if in a separate folder from the web application backups.

Release Notes 7.8.000010

Main Focus: Configuration file support for Service Accounts

- Configuration files can now be managed for Service Accounts.
  - Secret Server can update hardcoded values stored in configuration files using Regular Expressions when changing service account passwords.
- Secret Dependency Page updated to more easily handle ordering (drag and drop) and Dependency specific information.
- Added Active Directory synchronization optimizations for large domains.
- New Folders default to inherit permissions.
- Added Group handling to Advanced XML Import.
- Diagnostics page now includes database name for configuration purposes.
- Secret Template edit automatically re-focuses to next row when adding fields.

Bug Fixes

- Fixed XSS vulnerability with the privileged account picker control.
- Fixed open redirect vulnerability on the Login page when already logged in.

- Fixed possible database connection error for long running Active Directory synchronizations and other background threads.
- Fixed auto complete issue on some sensitive fields.
- Heartbeat status is now automatically updated when RPC succeeds.
- Fixed issue with Oracle password changing failing on passwords with certain special characters.
- Fixed issue with Agents not properly failing over in clustered instances.
- Fixed issues in advanced XML import when loading items with duplicate permissions.
- Fixed issue with incorrect lockout warning on Group and Role Assignment page.
- Fixed error for Event Subscriptions with inactive users.
- Fixed potential timeout errors on Diagnostics page.

Release Notes 7.8.000002

Bug Fixes

- Fixed issue with web services for Windows Authentication not enabling properly.

Release Notes 7.8.000001

Bug Fixes

- Fixed wording of confusing instruction text when changing a Secret's Template.
- Fixed header version to reflect the correct version.

Release Notes 7.8.000000

Main Focus: Password Changing Integrations and Custom Launchers

- Created Java API for use in embedded scripts without hardcoding a password.
  - Examples: Java API Examples KB
  - Deployment instructions : Java Console Instructions KB
- Added MySQL Password Changer and Template.
- Added OpenLDAP Password Changer and Template.
- Added DSEE Password Changer.
- SQL Server password changes can now use a privileged account.
- Admins can now create configurable LDAP based Password Changers.
- Added Custom Process Launchers to start user specified applications on a client machine with credentials from the Secret.
  - Added PowerShell, SQL Management Studio, and Sybase iSQL custom launchers.
- Added XML Export option to simplify restoring or migrating from an export.

- Added support for sys accounts for Oracle password changes."
- Updated Activation to handle VM environments better.
- Added Convert Secret Template.
- Added option to Check Out a Secret without changing the password on Check In.
- Added new report to show Secrets with pending approval requests.
- Added change password web service method. Bug Fixes
- Fixed bug where disabled accounts in Active Directory did not get automatically disabled in Secret Server.
- Fixed bug with dependency finder when using Agent.
- Fixed issues with Oracle connection strings exceeding allowed length.
- Fixed bug with Login Other Location in Firefox.
- Fixed bug with Secret Server

Release Notes 7.7.000012

Main Focus: Secret Server Installer Improvements

- Added MSI for initially installing Secret Server.
- Added ability to create the database if it does not exist during installation.
- Added support for a RADIUS failover server.
- Added more descriptive message when secret is checked out and then accessed from mobile devices.
- Added message to Role page to highlight any permissions that are not currently assigned.

Bug Fixes

- Fixed bug with visual keyboard that caused it to not submit correctly.
- Fixed bug where error occurred when using Unlimited Administrator and attempting to checkout a Secret.

Release Notes 7.7.000009

Main Focus: Secret Template Improvements

- Added auditing to all Secret Template and Secret Field actions.
- Updated Secret Fields to use a soft-delete so the data can be retrieved.
- Added Chrome support for Copy-to-Clipboard.
- Added clustering support for Remote Password Changing Agents.
- Added embedded searching and Page Size settings to most Admin Logs and Grids.
- Added exception logging to SQL Account Password Changing.

Bug Fixes

- Fixed issue with Expired Secrets not sending event alerts.
- Security Fix for restricting the search textboxes to a max length.
- Security Fix to prevent XPath expressions with the language resources.

Release Notes 7.7.000002

Enhancements

- Created the Password Compliance Report Category.
- Renamed the Non-Alphanumeric Character Set to Symbol.

Bug Fixes

- Fixed bug where the Remote Desktop Launcher was not properly cleaning up configuration files.
- Updated the Password Requirement edit page to prevent overriding the minimum length while entering the maximum length.

Release Notes 7.7.000001

Bug Fix

- Fixed bug where Secret Update email alerts are triggered by checking Password Compliance. Release Notes 7.7.000000

Main Focus: Advanced Password Requirements

- Advanced rules can now be applied to password fields on the Secret Template.
  - Multiple custom character sets can be created and used in these rules to more exactly limit the type of password generated.
  - New reports to show what passwords do not meet complexity requirements.
  - Validation can be enabled to prevent saving Secrets that do not meet the password complexity requirements.
- Added audit record for machine when using an Active Directory account to launch Remote Desktop and PuTTY.
- The advanced XML import now includes Secret dependencies. Bug Fixes
- Fixed bug in the color column on custom reports.
- Fixed bug that could cause the Local Account Finder in Discovery to fail for some sets of credentials.
- Fixed bug where the default folder was not always being set on Dashboard.

Release Notes 7.6.000000

Main Focus: Discovery

- Discovery: Account Import (Enterprise Plus)
  - Administrators can now scan for domain joined machines and import local Windows accounts into Secret Server.
- Dependency Ordering
  - Dependencies can now be ordered and a wait time can be specified which will be observed before the Dependency is updated.
- Added new Password Changers for Juniper, HP ILO, and Blue Coat Devices.
- Added option on custom password changers to specify line ending type (CR/LF).
- Added new Web Services methods for file upload and download from Secrets.
- Added new Bulk Operation to set the privileged account for Windows and AD Secrets.
- Added Secret Copy event for use in Event Subscriptions.
- Added configuration option to send Syslog/CEF messages by TCP instead of UDP.

Bug Fixes

- Fixed bug where Secret Copy created an Edit Audit Record.
- Fixed bug where dates in reports did not observe the user's date format preference.
- Fixed bug with dates as report parameters on non-US SQL installations.
- Fixed bug where unchecking All on Secret Template History caused error.

Release Notes 7.5.000002

Bug Fixes

- Fixed cross-site scripting (XSS) vulnerability on Secret View screen related to URL fields.
- Fixed command injection vulnerability in the PuTTY Launcher.
  - (These were reported by a customer performing a security audit – the issues were confirmed, fixed and released within 24 hours by the Secret Server team.)
- Fixed issue with limited number of concurrent Agents being able to connect.

Release Notes 7.5.000001

Bug Fixes

- Fixed Configuration page to only show video codec option when Session Recording is on.
- Fixed bug where Secret Server uses excessive CPU resources related to new Discovery capabilities.

Release Notes 7.5.000000

Main Focus: Discovery and Session Recording

- New Discovery Network View (Enterprise Plus)
  - Brings together the view of the network and the Secret Server repository to show Administrators whether local accounts on Domain Computers have corresponding Secrets.
- Session Recording (Enterprise Plus)
  - Remote Desktop or PuTTY sessions can now be recorded and the full movie is available as part of the audit. This setting can be configured per Secret and role permissions control who can access the audit movie.
- Hide Launcher Password setting can now be configured per Secret as an alternative to the role permission.
- Users are now automatically redirected from the pending request page when their request for access has been approved.

Bug Fixes

- Fixed copy to clipboard bug in Remote Desktop launcher.
- Fixed bug where users were not correctly removed from Groups in Secret Server during synchronization when the AD Group is empty.
- Fixed bug where CEF port defaulted to -1 in Configuration.

Release Notes 7.4.000002

Bug Fixes

- Fixed bug in Approval for Access Quick Pick control.

Release Notes 7.4.000000

Features and Enhancements

- New Enterprise Plus Edition
  - Added SIEM integration using CEF and Syslog formats.
  - Support for front end server clustering.
- Added Group filter on Active Directory Synchronization screen.
- New Copy Secret option.
- New Delete Secret Role Permission.
- New Events for Users.
  - Login, Logout, Login Failure, and Password Change
- File attachments are now stored in the database rather than the file system.
- Added new Advanced Import option from XML. Bug Fixes

- Calendar on Approve Access now respects all date formats.
- Fixed Tab and Copy to Clipboard bugs in IE9.
- Fixed issue where users assigning groups needed Administer Roles permission.
- Search box on Dashboard is now automatically given focus.
- Fixed bug with Secret data not always formatting correctly in Dashboard Widgets.
- Fixed bug where option to view deleted secrets showed incorrectly on Dashboard.
- Fixed bug with single quote in search breaking not working on dashboard.
- Fixed security issue with Ajax services.
- Fixed bug with alternative Active Directory account name formats not being supported.

Release Notes 7.3.000002

Security Update

- Fixed potential cross-site scripting vulnerabilities on Administration screens. (This was reported by a customer who performed a security audit – the issue was fixed and released within 24 hours by the Secret Server team.)
- View this Knowledge Base article for having Secret Server require secure cookies. This is done through changing a setting in the web.config.

Release Notes 7.3.000001

Main Focus – Bug Fixes

Features and Enhancements:

- Updated the Browse widget on Dashboard to highlight the search term when the tab loads.
- Added Activate Offline button. Bug Fixes:
- Updated License Activation to support Unicode characters in the license name.
- Fixed bug in the phonetic icon on Secret View. 7.3 Main Focus – User Interface Improvements Features and Enhancements:
- Added a new front end home page called Dashboard. For a movie preview click here
  - Multiple Customizable Tabs.
  - Dragable Widgets.
  - Report Widgets.
  - Expandable Secret View in search results.
  - Streamlined Folder and Secret search.
- Added new setting for how unmasking a password works (hold versus single click).
- Added new header menu with drop down navigation.
- Added additional auditing to the upgrade process.

- Added license activation to Secret Server, existing customers have 30 days to activate. Bug Fixes:
- Fixed bug in DBConnectionReset page.
- Fixed bug in Users Activity Report.
- Fixed bug where the application would sometimes give an error after a fresh install.
- Fixed validation bug in assigning Role by User.
- Fixed bug in Dependency finder where unchecking the 'select all' did not unselect all computers.
- Fixed bug in Search having to do with inactive groups.
- Extended RADIUS two factor timeout. 7.2.000003 Features and Enhancements:
- Added Folder Path, whether child folders were exported, and number of secrets exported to Export Log grid.
- Added audit records to each secret when exported. 7.2.000002

Bug Fixes

- Fixed issue in 7.2.000001 that could cause duplicate users to be created during an AD sync.

Release Notes 7.2.000001

Bug Fixes

- Fixed bug in Active Directory Synchronization for custom schemas.
- Fixed memory issue in Active Directory Synchronization for large domains.
- Fixed bug in the Event Engine administration section for Professional Edition.
- Fixed bug with two factor pin code email timeouts.

Release Notes 7.2

Main Focus: Event Subscriptions ("Custom Alerts") and Active Directory Synchronization Performance

Features and Enhancements

- Event Subscription feature:
  - Users can receive email alerts for custom event subscriptions.
  - Subscription events include: Unlimited Administration Mode toggle, Secret Edit/Add/View, Role and Group Assignment changes, Secret Expiration, Configuration changes, and many more.
- Improved Active Directory Synchronization to reduce time spent retrieving domain information.
- Added option to additionally force owners and approvers to request access on a Secret.
- When approving access to a Secret, users can specify the access window down to the minute.
- Added optional port field to the default Oracle Template and Oracle Remote Password Changer.
- Increased performance for folder permission updates.
- Removed Security Code from Credit Card Template for new installations for PCI compliance.

Bug Fixes

- Fixed bug where duplicate Secrets could occur during create.
- Fixed bug with assigning groups by users for administrator role validation.
- Fixed bug where the custom command test action did not correctly replace all parameters.
- Updated Heartbeat to perform additional validation in cases where accounts may not have the login privilege.
- Fixed web launcher for Chrome and Safari.

Release Notes 7.1.000015

Main Focus: Usability and Performance

> NOTE: An important security update has been released for the Microsoft .NET Framework. Please ensure that this update is installed on your server to ensure maximum security. For further detail and how to obtain the patch, please click here.
>
> NOTE: We are phasing out support for Microsoft SQL Server 2000. Future releases will not support Microsoft SQL Server 2000. Features and Enhancements:

- Added support for changing Scheduled Tasks on Windows Server 2008 and Windows 7 instances.
- Improved Search performance for highly nested folders.
- Offline upgrades can now be performed by uploading a local zip file.
- Database Connection Reset page now resets the application automatically.
- Require Comment to View and Approval for Access can now be applied to the same Secret.
- Require Comment to View coincides with checking out a Secret.
- Secret Access Request now shows full request history on Pending Requests page.
- Notification emails sent for Request Reason now contain the user entered reason comment.
- Added ability to encrypt the instance encryption key with DPAPI for added security.
- Backup file path now allows all valid special characters.
- Allow setting an AutoChange schedule on a Secret before enabling AutoChange. Bug Fixes
- Fixed bug in Integrated Authentication with local Windows Accounts.
- Secrets mapped to Users through inactive groups are no longer visible in custom reports.
- Fixed exception that occurs on Remote Password Changing Agents after upgrades.
- Fixed issue where updating file attachment did not save in certain situations.
- The Enter key now works on home page search box.
- Fixed sort for inactive users on User Administration page.
- Fixed Active Directory Synchronization login error on Domain search when fully qualified username was not used.
- Fixed bug in Oracle password changing by updating template to allow additional parameter specifications.

- Fixed bug in the autopopulate search where clicking a Secret failed to navigate to the Secret view page.
- Fixed error when running "Test Action" on remote password changer custom commands. 7.1.000001 Security Update
- Updated Error Reporting in order to address a vulnerability in ASP.Net. For more information see this Knowledge Base article

Release Notes 7.1.000000

Secret Server Agent

- Use Remote Password Changing, Heartbeat, Dependency Finder on external networks.
- Easy Agent Installation with MSI.
- High Security: Full over-the-wire Encryption.
- Requires no incoming ports on the Agent network.
- Customizable URL and Server Port.
- Light-weight bandwidth usage.
- Client automatically upgrades when Server is upgraded. Require Comment
- Require Comment when a Secret is Viewed (useful for tracking change control numbers).
- Bulk Operation to enable Require Comment on Secrets. More
- Major Database Performance increases in Home, Secret View, and background threads.
- Added Bulk Operation for Remotely Changing the Password. This can be used to keep multiple accounts in-sync with the same password.
- Improved Search in Navigation Bar to go directly to the selected Secret (when unique name).

Bugs

- Updated Search Indexer to run as a batch process.
- Fixed performance issue when Unlimited Administration Mode is turned on.
- Fixed RDP Launcher to work consistently for local Windows Accounts on Windows XP machines.

Release Notes 7.0.000040

Remote Password Changing

- Added Cisco password changing support (SSH and legacy Telnet).
- Added Unix Root Account password changing using separate Secret for login.
- Added the Remote Password Changing tab for configuring options on a Secret (moved AutoChange checkbox to this tab).
- Password change can be set up for Active Directory and Windows accounts using a privileged account instead of the account changing its own password.

- Added the ability to create configurable command sets for handling different platforms and operating systems to do password reset using SSH or Telnet (including using credentials from other Secrets).
- Added the ability to test Password Reset and Verify from an admin dialog.
- Added the ability to specify the port for password changes when using SSH and Telnet.
- Added button to allow cancellation of Change Password Remotely.

Heartbeat

- Secret Heartbeat will test the credentials stored in Secret Server on a periodic basis to ensure they are still valid.
- Receive email alerts when a Secret fails the Heartbeat.
- Supports all Remote Password Changing templates and Password Verify. Web Launcher
- Web Launcher to automatically login to websites using credentials stored in Secret Server.
- Web Launcher bookmarklet for single click login from the browser (supports all browsers).
- Note: Secret Assistant is being retired in favor of the Web Launcher and bookmarklet (Secret Assistant is still supported but no longer recommended).
- Automatic download option for the latest Web Launcher settings for commonly used sites from #{COMPANY}#.com. Search
- Made extended Search Indexer split indexed terms into 3-12 character segments instead of just 3 character segments.
- Made extended Search Indexer not split the search term before searching.
- Improved order of search results. Exact matches on name will be on the top, followed by 'like' matches in the name (ordered by name) and then secret item hash matches (ordered by name). More
- Added webservice to use Integrated Windows Authentication to allow scripts to run without having embedded username/password and retrieve passwords from Secret Server.
- Updated Active Directory synchronization to support Child, Parent, and Sibling Domain Credentials.
- Changed all random number generation to use System.Cryptography.RandomNumberGenerator for improved security.
- Increased the hash iterations on both local user passwords and DoubleLock passwords to provide additional security against brute force attacks on the hashes.
- Extended IP Address Range restrictions to work for class A and B networks.
- Added Maximum Offline Minutes feature so that mobile devices can only cache data for a limited time.
- Added a Generate Password button to the "Change Password Remotely" page.
- Split Unlimited Administrator role into "Administer Unlimited Admin Configuration", "Unlimited Administrator", and "View Unlimited Admin Configuration".
- Changed minimizing on Copy to Clipboard to be a per user preference.

Bug Fixes

- Fixed "No process is at end of pipe" SQL exception that occasionally occurred after doing an iisreset.
- Added email addresses to all users during Active Directory synchronization even if disabled in Secret Server.
- Fixed URL field on Secret to open correctly if http:// is not included.
- Fixed SSH issues when changing passwords on SUSE Linux.
- Fixed the ActivityDirectorySynchronization page, the AvailableGroups listbox no longer displays Groups that have been removed in AD.
- Added saving of the ADGuid for new groups when Save button clicked on the Group Synchronization page (instead of waiting for first AD sync).

Release Notes 7.0.000001

Features and Enhancements

- Added the ability to specify the characters to separate on when building the Search Index. Note: On upgrade the current search index will be rebuilt.
- Updated Dependency Finder to allow the user to manually specify the machine names to search.
- Disabled the trace and debug settings from the Web.config by default.
- BUG: Fixed Administration Export for IE when SSL is enabled.
- BUG: For XP machines, fixed the unsupported hash algorithm error for both the Email Pincode process and the Search Indexer.
- BUG: Updated RADIUS login to process passwords greater than 16 characters long to support Yubikeys.

Release Notes 7.0.000000

Main Focus: Custom reports, support for RADIUS, and more

Features and Enhancements

- Add-ons are now Professional and Enterprise Editions (explain Editions)
- Reporting
  - Reports page allows administrators to view standard reports, or to create reports with SQL and charting options. Reports can use a variety of 2D or 3D charts.
  - Reports can be displayed with all their associated data points (grid).
  - Reports can be placed into categories, and these categories and their reports can be organized using drag and drop.
  - Reports can have rows with different colors based on data values
  - Reports can be created using parameters such as start date, end date, and user ID.
- Added support for RADIUS integration to authenticate to Secret Server. This will work with AuthAnvil tokens, RSA tokens, and any other authentication scheme that supports RADIUS.

- Secret Server now uses FIPS 140 compliant algorithms and operates normally when limited to FIPS 140 only under Windows Security/Group Policy.
- Auto-complete added to Secret search textbox.
- Terminology change – renamed "inactive" to "deleted" for Secrets.
- Added scrollbars to Search and Browse tabs in homepage – makes it easier when you have lots of folders.
- Added icons to permission grids to indicate person or group.
- Groups in permission grid are clickable, which shows the list of users in the group.
- Date time picker works with the user's preferred date/time format.
- Added "copy to clipboard" support for Chrome and Safari.
- The layout of the Configuration page is now categorized into tabs for better organization.
- Added IP address logging for all failed authentication attempts. Previously, only attempts that caused lockouts were logged.
- Improved localization so that messages that do not exist in the localized XML file are rendered as "Resource Not Found:".
- Changed the inactivity timeout timer to reset on partial postbacks. This means that users will not get redirected due to inactivity when browsing folders or searching for secrets on the home page.
- Added on-screen notification for support license expiration.
- Added Configuration settings for an instance level default Time and Date format.
- Added separate page (DBConnectionReset.aspx) to allow users to change their database connection information without going through the installer.
- Added the ability to reset a forgotten DoubleLock password.
- Added Folder Search to the Folder picker.
- Added Folder Templates to support Folder (default), Customer, and Computer.
- Greatly improved Home page performance for running BulkOperations for larger instances.
- Improved the Change Password screen to give instructions for the password complexity guidelines.
- Improved System log to support having a maximum number of rows and to alert administrators when the log is truncated (by 50%).
- Updated the Launcher to support having a "blank" domain for local accounts.
- Updated the Launcher to support credentials for launching into multiple hosts. The user will be prompted to enter the Machine or Host before the RDP or Putty instance is opened when wired to the "user input" field.
- Added a User and Group picker to replace the dropdown list for user and group assignment for large instances.
- Updated the User create process to automatically assign the "User" Role by default.
- Added a grid of the user's Roles on the user view page.
- Webservice additions and updates:

  - Added FolderId to the Secret get methods
  - Added the ability to specific the folder on Secret Create and Update ▪Added Folder webservices for Get, Create, Update, and Search
- Added support for RPC support for Sybase databases.
- Added the ability to migrate a local user to an Active Directory user and maintain the existing groups and permissions.
- Added the full Folder Path on the folder edit and create pages.
- Search Indexer will split by newline.
- Added icon for NATO phonetics translation of Secret field on Secret View page for reading information verbally.
- Added Login form to the "Logged in at another location" page.
- Update the Resource Provider to support changing a single element with custom resource such as the Help link.
- Session Timeout has been moved to external config file to prevent overriding settings on upgrade.
- Added folder picker and "include subfolders" option to the User Audit report.
- Added "Last Date" column to the user audit report page.
- Added "Save to File" functionality for many grids.
- Added common table expression functionality to folder database queries to improve performance on SQL Server 2005 and SQL Server 2008.
- Updated code signing certificate for Launcher.

Bug Fixes

- Fixed bug that caused Dependency Finder to time out prematurely for some systems.
- Changed "lock out" for Web Services to be consistent with logging in through the Web interface.
- Removed unnecessary validation when entering a new domain that required the domain account to have reset password permissions.
- Fixed issues with Admin Secret Export for some browsers.
- Fixed Dependency to show all computers found in Active Directory.
- Fixed the Keep Alive thread and other background threads to avoid spamming the system log when thread cannot be stopped.
- Fixed the Active Directory Group Synchronization page to display the listboxes with a proper width for all Browsers.
- Expanded the SQL timeout on backups to support large instances.
- Updated Active Directory synchronization to properly assign membership for groups made up of both child and parent domain users.
- Fixed the display of login policy to fit inside the box.
- Turned off autocomplete for password textboxes on the "Secret Edit" screen.

Secret Server Release Notes 6.x

Release 6.2.000013

Main Focus: Bug fixes

- Fixed bug where Folders would not be visible in Unlimited Admin Mode.
- Fixed bug when adding a new domain with a non-Administrator account.
- Fixed bug that caused Active Directory synchronization to crash if an AD user could not be accessed.
- Fixed bug that would incorrectly enable an AD user that exists in AD and Secret Server but are not in a synch group.
- Fixed bug related to Remember Me value and Inactivity Timeout.

Release 6.2.000012

Main Focus: Responding to customer requests

- Added support for child domain users being members in parent domain groups.
- Remote Desktop Preferences for the Launcher o Copy to clipboard, admin/console, attach drives, share printers
- Ability to Delete IP Address Ranges
- Embedded mode to Hide Headers and Footers o Running Secret Server in Embedded Mode KB
- Improved support for Database access through Windows Authentication to have the background thread run with identity of the site instead of AppPool
- Added Permission and confirmation for force expiring secrets on the User Audit Report.
- Added Full Path to folder in Secret View and Edit alerts.
- Improved the performance on the Domain Synchronization for selecting AD groups.
- Made Favorites click through to its own bookmarkable page.
- Terminology Change: "Owner" permission replaces "Share."
- Improved and fixed bugs in Backup: o Backup respects setting for not sending failure emails to Administrators o Fix scheduled backup inconsistencies for some users o Limited to 3 retries
- Added better support for incomplete language files, so defaults to English if item is not found.
- Increased folder performance for renaming and editing permissions.
- Updated Domain Synchronization to set the DisplayName for new users and support username changes in Active Directory.
- Updated display issues with listboxes being too small on the Group Edit page and Domain Synchronization page.

Release 6.2.000006

- Fixed bug with the Role Assignment screen showing duplicate groups.
- Fixed bug where the Everyone group was not appearing in the Group assignment dropdown list on the permission screens.

Release 6.2.000005

Main Focus: Remote Password Changing enhancements and performance tuning

Features and Enhancements

- Disabled autocomplete on the Next Password textbox for Remote Password Changing.
- Service account credentials in these formats are now found by the dependency finder:
  - o username@fulldomainname
  - o username@shortdomainname
  - o shortdomainname\username
  - o fulldomainname\username
- Updated the Expired Secret log to include when the Secret is not changed due to the expiration time schedule.
- Performance improvements when using Unlimited Administrator Mode.
- Performance improvements on the Folder edit page.

Bug fixes

- Remote Password Changing will no longer fail when a privileged account on a dependency is not set. Instead, it will attempt to use the credentials on the Secret.

Release 6.2.000004

- Fixed minor bug that incorrectly displayed encrypted values after saving a Secret.

Release 6.2.000003

Main Focus: Usability and Workflow

Features and Enhancements

- Streamlined the Secret creation process
  - o Single click for folder selection
  - o Remembers last selected folder o Allow changing Secret Template on the Create page
  - o Combined Search and Browse last selected Folder
- Option to allow Secrets to require approval for access o Email Notifications to approvers and requestors o Audit is kept of all approve and deny actions o Secret Access Request Manager page

Bug fixes

- Fixed the missing folder indentation in IE

Release 6.0 6.2.000000

Main Focus: Responding to customer requests

Features and Enhancements

- Users can now reset their login password through a password reset email.
- Added configuration option to AD synchronization to prevent enabling and disabling users during synchronization.
- Added ability to synchronize email addresses for AD users.
- Added "LockedOut" feature so that failed authentication attempts locks out a user instead of disabling them.
- Added ability to specify whether or not Windows Service dependencies should restart after a password is changed remotely.
- Added ability to handle AD hierarchies that contain cycles in their groups.
- Added several new webservice methods to support the new Secret Server iPhone application.
- Added a password migration tool for Password Corral (See the Tools page in Secret Server for more details).
- Added option to enable a Keep Alive thread so that the ASP.NET worker process never gets shut down.
- Added an audit record for when the launcher is used.

Bug fixes

- Fixed bug where inactivity timeout did not work correctly.
- Fixed bug that allowed users to delete folders containing Secrets when the "Require folder for Secret" option was turned on.
- Fixed bug where Windows Integrated Authentication through AD did not work for domains not hosting.
- Fixed bug where some AD hierarchies that had root folders with no users in them could cause null reference exceptions.
- Fixed bug where JavaScript was not getting cleared from cache on upgrades.
- Fixed bug that allowed users to view folders and their audits without the appropriate permission setting.
- Fixed bug where a Secret could be created from an inactive Secret Template if the query string was entered.
- Fixed webservices to observe IP address restrictions.
- Fixed bug where inactive roles were being displayed on Admin Role Assignment pages.

Release 6.1.000002

Main Focus: Minor updates to 6.1

Features and Enhancements

- Introduced the Failover Partner on Step 3 of the installer to support mirrored database environments.
- Added the use of the legacy Search / Browse functionality before 6.1 as a preference.
- Added an option to allow Browse to also include the subfolders.
- Added a Diagnostics page to assist troubleshooting.

Bug fixes

- Fixed bug where certain operating system settings would prevent users from being able to create a DoubleLock password.
- Fixed bug where the Launcher application did not start correctly.
- Fixed bug where URLs contained in email alerts did not contain the right link.
- Fixed link to a Knowledge Base article on the Backup Configuration page due to KB article restructuring.
- Fixed minor security issue where creating a user with a special sequence of characters would cause unexpected behavior.

Release 6.1.000000

Main Focus: DoubleLock for sensitive Secrets and bug fixes

Features and Enhancements

- Implemented DoubleLock to provide an additional security layer for sensitive Secrets
- Enhanced performance for Active Directory authentication
- Separated the "Search" and "Browse" functions on the Home screen
- HTML now renders using "standards mode" (may affect user customized themes)

Bug Fixes

- Passwords generated for expired Secrets now meet domain credential requirements
- Fixed bug pertaining to an infinite redirect loop related to session expiration and password expiration
- Fixed bug where exception occurred on SecretGet webmethod when user has no permission to a particular secret
- Fixed bug with bulk operations where progress was not reported to the user
- Fixed bug where file attachments with spaces in their names didn't download properly
- Fixed bug where folder name appeared outside of the dialog when viewing a folder
- Fixed bug where multiple PIN codes were sometimes sent when using Windows Integrated Authentication
- Fixed bug to not allow Checkout to be enabled when Remote Password Changing is disabled
- Fixed broken Upgrade link in Firefox

- Fixed bug where users with permanent cookies disabled were always redirected to LogoutAnotherLocation screen
- Fixed bug to prevent users disabling Autochange on Secrets that require Checkout
- Fixed bug where IOException was occasionally thrown during installation due to file permissions
- Fixed bug in client-side JavaScript on installer
- Fixed bug that caused NullReferenceException when inactivating a Secret without the required role permission
- Fixed bug that occurred in user auditing when using an IPv6 address
- Fixed UI layout on the dependencies tab related to the explain link
- Fixed bug on Minimum Password Age validation when all fields are zero and checkbox is unchecked
- Fixed bug when unmasking passwords that have XML special characters 6.0.000001 Main Focus: Minor Updates to 6.0 Features and Enhancements
- Added support for encrypted connections to SQL Server.
- Changed installer to not overwrite customized configuration files in future releases.
- Extended password length to 127 characters on AD credential used for AD Synchronization. Bug fixes
- Fixed bug where expired password and expired license caused redirects.
- Fixed bug where user with an expired local password could still use webservices.
- Improved stability of AD Synchronization capabilities.

Release 6.0.000000

Main Focus: Remote Password Changing and user experience

Features and Enhancements

- Enhanced Remote Password Changing to allow setting a specific date and time schedule for changing service account passwords and their dependencies.
- Dependent Windows Services are now automatically restarted when a service account credential is changed.
- Added Remote Password Changing support for Oracle accounts.
- Users can now specify their preferred date/time format.
- Added new role permission to use the launcher feature without being able to view the password on the Secret.
- Added AJAX support to various features to enhance the user experience.
- Disabled the 'Search by Active Secrets' option for users without the 'View Inactive Secrets' permission.
- Improved performance of initial AD sync page load.
- Updated Russian Localization to support new features.

Bug fixes

- Fixed bug where content was not correctly displayed on the 'Expired Secret' report page.
- Fixed intermittent JavaScript error related to the scroll position on pages. Compatibility:

- Secret Server 6.0 no longer supports Windows 2000 due to our upgrade to the Microsoft .NET Framework 3.5.

Secret Server Release Notes 5.x

Release 5.1.000001

Minor Updates to 5.1

- Changed link on Administration pages, from "Languages" to "Language Maintenance"
- BUG: Fixed issues with URL case sensitive localization causing mixed languages to be displayed.

Release 5.1.000000

Main Focus: New email alerts and support for PuTTY Features and Enhancements

- Added support for launching PuTTY for UNIX-based secrets
- Added ability to receive email alerts when secrets are viewed
- Added ability to receive email alerts when a dependency fails to update on an automatic password change
- Added new role permission for searching/viewing inactive secrets
- Changed folder creation/movement to only require edit permissions on the parent folder
- Added support for Remote Desktop launcher with Windows Integrated Authentication
- Added new bulk operations for deactivating and setting autochange on secrets
- All pages now maintain scroll position on postback
- Added a Languages page for Administrators to update and translate content to their language of choice
- Added an OK button to the top of the Folder picker
- Added additional folder management buttons to the top of the Folder Administration screen
- Added functionality to make Secret Server 64 bit compatible
- Searching on all fields no longer splits words up by periods Bug fixes
- Fixed bug on Login where a minimum password age error was shown when creating a local user
- Fixed bug with Windows Service Dependency Changers when using Windows Accounts due to a missing prefix of the machine name
- Fixed bug related to unlimited setting on Remember Me
- Fixed null reference bug on Secret Audit when user does have "View Secret" role permission
- Fixed bug where an incorrect validation message was displayed when password history was set to 'all'

Release 5.0.000002

Main Focus: Minor enhancements to 5.0

- Improved database indexes for search functionality.
- BUG: Fixed issue that intermittently occurred in older Secret Server instances when upgrading.

- BUG: Fixed to not send alerts when search indexing.
- BUG: Fixed Secret Template to not allow search indexing on file attachments.
- Fix: Cleaned up the CSS and layout on several pages.

Release 5.0.00000

Main Focus: Changing Passwords for Scheduled Tasks and Service Accounts Features and Enhancements:

- Enhanced Remote Password Changing to update dependent Scheduled Tasks, IIS AppPools and Windows Services.
- Added Checkout option to provide accountability for the use of a secret – the password gets changed automatically on checking.
- Enhanced search functionality to allow users to search by all fields.
- Implemented 'Change Password Remotely' feature to allow users to immediately change a password on a remote server.
- Added new default theme to enhance the readability of the UI.
- Export by folder now includes all child folders.
- Added the SecretID field to Secret Server webservices to provide integration for custom development.
- Administrators can now force local user password expiration.
- Added configurable minimum password age requirements for local user passwords.
- Added password history configuration options to prevent users from using past local user passwords.
- Webservices and Secret Assistant usage now creates view audit records.
- SSH Remote Password Changing now works for "root" accounts.
- Added ability to automatically delete excess database backups on the application server. Bug fixes
- Fixed bug that occurred when trying to access the Administer Groups page with no active local groups.
- Fixed unlimited remember me bug with Secret Assistant.
- Fixed bug when trying to create a new secret from a Secret Template with no fields.
- Fixed bug where SSH remote password changing left open connections.
- Fixed bug where Secret Assistant would return inactive secrets.

Secret Server Release Notes 4.x

Release 4.3.000000

- Implemented 'SH for password changing on Linux accounts.
- Fixed bug with Active Directory Synchronization when pulling users and groups from an organizational unit.
- Fixed issue with the 'next password' component of Remote Password Changing. 4.2.000000 Main Focus: Enhancing Folder Functionality and Security Features and Enhancements:
- Added configuration option to allow Secrets to inherit folder permissions by default.

- Added configuration option so that a user must have view permission on a folder to see it.
- Users can now create and manage their own folders without them being visible to all users.
- User now requires Edit permission on a folder to be able to add secrets to it.
- Added a new 'Everyone' group to include each existing user for easier management and legacy folder permission support.
- Tightened folder restrictions to require share permission on a parent folder in order to add a child folder.
- Implemented audit records for when Groups are created, made inactive/active within Secret Server.
- Implemented audit records for when users and groups are created or made active/inactive from Active Directory.
- Renamed two Role Based Security permissions: Administer Roles is now Administer Role Permissions and Administer Group Roles is now Administer Role Assignment.
- Secret Types are now labeled as Secret Templates.
- Added an 'Evaluation Expiry' notice to alert users when their evaluation is about to expire. Bugs:
- Fixed bug when users were made inactive when Secret Server could not connect to Active Directory.
- Fixed bug where Backup did not work properly if a database name contained certain characters.
- Fixed error that occurred on the AdminGroupByGroup page when no groups exist.
- Fixed error when trying to import folders with line breaks in a Secret field.
- Fixed issue with Password Type configuration not saving correctly in certain situations.

Release 4.1.000000

Main Focus: Addressing Role Based Access Control Features and Enhancements

- Implemented Role Based Access Control (Role Based Security) to set granular, assignable permissions for users.
- Added the ability to launch Remote Desktop from a secret.
- Added the ability to import secrets by folder.
- Secrets can now be exported with a folder name.
- Added "Run Now" button to the Remote Password Changing screen.
- Implemented a visual keyboard on the login screen to thwart keyloggers.
- Added the ability to create custom web.config files to override the default impersonation settings that will not be overwritten on upgrades.
- Added a dropdown on the results screen for users to define the amount of secrets to display.
- Created a Security Hardening Report that displays the security level of your system's installation.
- Created the SecretTypeSetActive.aspx page for quickly setting the active status on Secret Types.
- Improved the "Help" documentation.
- Groups deleted from Active Directory will now be disabled.

- Improved performance by adding caching for theming.
- Specific passwords can be set on the Remote Password Changing – AutoChange feature.
- Added a preference for showing a full folder path on the home search grid.
- Implemented robot.txt file to stop search engines from indexing Secret Server installations.
- Folder creation and editing is now an assignable permission.
- Added a search textbox to the Users screen.
- All cookies are now HTTP only for additional security.
- Added "Save and Add New" button SecretView.aspx.
- Increased the visual size of the notes field. Bug Fixes:
- Fixed bug where an exception was thrown when invalid information was entered in the "minimum password length" configuration option.
- Fixed bug where the folder picker modal did not work properly when Secret Server was viewed inside a frame.
- Fixed error where Secret Type export XML format was incorrect.
- Fixed bug where notification emails did not contain the full URL for the installation.
- Fixed bug where Integrated Authentication was not setting last login.
- Fixed bug where permission checkboxes were being displayed when the secret was set to inherit permissions from folder.
- Fixed bug where duplicate users appeared in the Active Directory synchronization preview.

Release 4.0.000003

Main Focus: Improving Permission Inheritance and Bug Fixes

Features and Enhancements

- Bulk operations now supports enabling folder inheritance on a secret.
- Deleted Synchronized Active Directory groups are now disabled within Secret Server.
- Added support for automatic backups on servers at different locations. Bug Fixes:
- Fixed bug when editing folder permissions that include a disabled user.
- Fixed padding error for secret item history for very large values on secrets.
- Fixed bug in Remote Password Changing due to new column for inherited permissions.
- Fixed broken "unmask password" image on 'Secret Edit' page.
- Fixed 'Remember Me' bug due to .NET 2.0 migration.
- Fixed 'Close' image on dialog.
- Fixed paging problem on AdminExport grid.
- Fixed bug where expiration date did not decrease on old secrets.

Cloud Archived

This section contains versions of Secret Server Cloud that are over a year old.

For changes pushed to cloud between major releases, see the Secret Server Change Log.

Secret Server Cloud Release Notes for June 3, 2023

Release Dates and Notes

Cloud: June 3, 2023

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.8.0

Protocol Handler: 6.0.3.26

New Features

Checked Out Secret View

Implemented the Checked Out Secret view to Quick Access in the Secrets Folder panel. This is a quick view, showing users all of the secrets that are currently checked out to them.

RADIUS Silent Answer

Silent Answer is a new RADIUS configuration option that allows setting the response to a predefined string value. This is to support push notification and other interactive variations in advanced RADIUS authentication configuration. The new setting replaces "Attempt User Password" and allows for sending the user password or another predefined string.

Check Out Recovery

We updated "Force Check In" to enable secret owners with the "Force Check In" role permission to force a check in of secrets that are configured to change their password upon check in. When a force check in is initiated, the secret is automatically checked out to the owner. This enhancement is particularly useful in scenarios where a secret is checked out with a failing RPC configuration and cannot be checked in by the current user. With this change, the owner can take over the checkout session, resolve the secret configuration issue, and then perform a standard secret check in.

Syslog Metadata for Launched Sessions

For built-in launchers, during a launch event, the launch target host is included within the details of the Syslog message as an additional "Host" field. Previously, this was only sent for launchers requiring host selection but now includes launchers with a static host-target mapping.

Launcher Administration Page Conversion

We updated the Launcher Administration pages under Secret Templates to use our new UI patterns with a modern design. No functionality is affected, but the page is more responsive and intuitive.

SSH Key Authentication Passphrase Requirement

We added a new configuration setting to the login configuration page that allows administrators to enable a mandatory requirement for passphrases when users generate SSH keys for SSH Terminal key authentication.

Enhancements

- Added a Managed field to the Discovery Network view to show when a discovery item is managed.
- Added a Password Age column to the redesigned Discovery Network View.
- Added a Quick Access link to see all Secrets you currently have checked out.
- Added filters to the Secret Search API endpoint to allow filtering results by checked-out status: *paging.filter.showSecretsCheckedOutByUser and paging.filter.showCheckedOutSecrets*
- Added info to logs to indicate why users cannot match or create users in SSC. Find this at Secrets > Admin > Platform Integration > Logs tab. Common notifications include DuplicateUserMappedToDifferentProviderName: The user was initially setup to a different Platform source, the URL or userid (provider key) changed, indicating the original use was deleted. MaxLicensedUsersException: All licenses are taken so additional users cannot be added.
- Added integration support for Platform users matching local SS users that do not have an @ in their name. If platform user is username@local or username@tenantname then the username portion will be used to match local users on the SS side.
- Added support for LDAP RFC2307 group membership, used in OpenLDAP.
- Added the option to require a passphrase for user public SSH keys.
- Added validation messages to password requirement rules for when password requirements are too complex to reliably generate a password.
- The Discovery Service Accounts detail page now displays both the services running under a directory account and the computers on which those services run.
- Distributed engines no longer need directory services enabled to perform discovery.
- Introduced a new Launch Secret role permission, which is needed to use launchers. This permission is automatically granted to roles with the View Secret permission, which previously controlled this behavior.
- Removed the secretitemvaluetransitionhistory.aspx page and replaced it with an API endpoint, removing the possibility of bypassing the Hide Launcher Password control.
- RPC heartbeat and password change logs are now full screen instead of a dialog box.
- The PowerShell script timeout no longer defaults to 90 seconds. Instead, it now uses the value from the Event Pipelines Maximum Script Run Time (Minutes) setting in advanced configuration.
- The new folder icon in the secret panel no longer shows if the user does not have the Administer Folders role permission.

- The user audit report now has a filter panel and a description for how rotated secrets are calculated for this report.
- There is now a pending RPC screen and a timer that checks you back in, blocking seeing secret info indefinitely.
- Users can no longer access secrets that have failed processing a password change. Instead, they are shown a message stating the change failed.
- We now initially load 60 secrets when viewing a grid to support 4k monitors. This was previously 30.
- Within the details of the syslog message, there is now a username field that contains the mapped username for the launcher on a launch event. It appears as Username: [<username>] for the built in launchers.
- Within the details of the syslog message, there is now a Host field with the value of the mapped host for the launcher on a launch event. It appears as Host: [<host>] for the built in launchers.

Bug Fixes

- Fixed an issue where Platform integration user synchronization failed if duplicate usernames existed in Secret Server.
- Fixed an issue where a secret template could be saved without RPC mappings configured.
- Fixed an issue where all event subscriptions did not fire for secrets in subfolders of the target folder.
- Fixed an issue where Disaster Recovery (DR) email alerts failed to send.
- Fixed an issue where extended fields were not properly exported to CSV files.
- Fixed an issue where keystroke data from the advanced session recording agent did not appear in the keystroke activity details area of the playback page.
- Fixed an issue where large messages from distributed engines to engine workers would not process. Engine workers may have crashed especially frequently in environments having four or more workers, including Secret Server Cloud.
- Fixed an issue where LDAP sync via distributed engines would not work when the base DN was different from DC.
- Fixed an issue where links on the Session Monitoring page while in grid mode would not correctly link to Secret Server Cloud with authentication.
- Fixed an issue where the API endpoint api/v1/secrets/{id}/fields/{slug}/ logged an audit that the password was displayed when the actual password was not returned to the user due to hide launcher password being enabled.
- Fixed an issue where the Confirm Action button in the bulk operation dialog box would remain active while the operation is processing. This is now correctly disabled to prevent initiating the action multiple times.
- Fixed an issue where the SubscriptionName condition for a notification rule would display the event subscription ID instead. It now correctly uses the name when the user has the appropriate roles to list the subscriptions.
- Fixed an issue where the terminate session mouseover tooltip displayed incorrect text.
- Fixed an issue with a secret template name validation message not showing.
- Fixed an issue with negative numbers exporting incorrectly when exporting a CSV.
- Fixed an issue with new Platform trials not creating personal folders in Secret Server.
- Fixed an issue with secret search producing SQL errors for customers with a lot of secret templates.

- Fixed an issue with stacked dialog boxes. The CSS styles for the Platform Opt In dialog box have been adjusted to align with Angular15.
- Fixed conditions that prevented users from being removed from a group due to the system incorrectly identifying that they would be unable to complete the same operation.
- Fixed issues with user and group syncing between Secret Server Cloud and Platform.
- Improved usability in specific UI areas to enhance the user experience.
- Updated `Createuser.aspx` to redirect users to the new User Management page.

Future and Recent Deprecations

This section describes planned future deprecation of feature or platform support in Secret Server.

Not applicable for the current release.

Secret Server Cloud Release Notes for May 6, 2023

Release Dates and Notes

Cloud: May 6, 2023

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.7.0

Protocol Handler: 6.0.3.26

Enhancements

Secret Folder Panel Redesign

We reworked the secret folder panel for additional functionality and a more streamlined user experience. Direct access to the folder panel from outside of the secrets view was found to unnecessarily clutter the navigation menu, and the panel is now visible on the Secrets page. In addition, we enhanced the following:

- We added a "Quick Access" section to the folder panel, which offers a single page combining the following sections:
  - Search
  - Favorite Secrets
  - Most Used Secrets
  - Session Secrets–Secrets accessed this session, allowing the user to return to a secret they have accessed in this browser session
  - Recent Secrets–Most recently accessed secrets, within this session or others
  - Shared With Me–Secrets that are shared with you but not in folders that you may view
- Added a "New Folder" button to the top of the folder tree.

- Pinned folders are now placed at the top of the tree instead of listed in a dropdown. Pinned folders give easier access to your favorite folders. When a pinned folder is selected, the displayed folder tree is based on the that pinned folder rather than the entire tree. The same applies to the content of "Quick Access," which displays secrets in the selected, pinned folder.
- A guide now displays for new users the first time they view the secrets page, introducing components and changes.

Other Enhancements

- DR: Created more robust data ambiguity handling when data replication processes a table where there is a multi-field unique key. These included giving precedence to source data when applicable or throwing an error when an ambiguity cannot be resolved.
- DR: Added an advanced configuration setting (defaulting to 3 hours) so that a long-running DR process will detect the configured amount of elapsed time and end the DR process, forcing the business user to manually run it again.
- DR: Read-only mode can now be enabled in Secret Server Cloud on the disaster recovery configuration page.
- Added a discovery import rule to the new network viewer.
- Added a link to the public SSH keys, when enabled, on both the user preference page and the administration tools section.
- Added a knowledge base link for Platform regions as part of the Platform Optin Experience.
- Added support for LDAP RFC2307 group membership, used in OpenLDAP.
- Discovery rules and dependencies grid can now be filtered by discovery source. Rule grid now also has discovery source available as a column.
- Local Admin column added to new discovery network view.
- Secret template name on the secret general tab is now a direct link to the template.
- The "Send Test Email" button now functions in read only mode.
- The report CSV download is now encoded so that certain Turkish characters appear, and the mime type was changed to text/csv.
- The unlimited admin page in configuration preview now has a link to open the unlimited admin audit.
- HSM Integration—RSA OAEP Padding Support. We added OAEP padding as a new configurable option when enabling or rotating the HSM integration. This is in anticipation of the planned deprecation of PKCS padding by NIST. Current configurations are unaffected, but this option is available when rotating the HSM key, or configuring a new integration.

Bug Fixes

Disaster Recovery

- Fixed an issue where DR email alerts were not being sent out.
- Fixed issues for password-requirement character-set data replication in the DR feature.

- Fixed an issue with disaster recovery replication where replicated custom launchers were not visible on their associated secrets.
- Fixed an issue with the disaster recovery logging process so that only error-free data replications are marked as successful.
- Fixed issue when replicating data for disaster recovery where pre-existing users on the replica that do not exist on the source could lose their All Vault Users group membership.
- Fixed replication to allow duplicate names to be replicated individually during disaster recovery. Groups with the same name will still be consolidated during replication when they share values for AD Guid, IsPersonal, IsPlatform, and DomainId.
- Password requirements are now replicated from source to replica as part of disaster recovery.

Other Bug Fixes

- Corrected logic that allowed password requirement consumers to bypass non-replicated secrets
- Dates in the report export no longer include the "Z" for UTC when server time is used and ISO date format is selected because the date/time is the server configured time and not necessarily UTC. That is, the date is ISO format and the server-configured time but does not include the offset. In some specific configurations when user format was selected, the timezone offset would be applied based on the actual server timezone and not the configured timezone.
- Discovery-specific OUs now returns results when the page is initially loaded.
- Due to security reasons, we removed the GET endpoint for /secretserversettings/export and replacing it with a POST endpoint where we can transmit the password securely. The contents of the payload are the same, except for new "password" and "doubleLockPassword" fields, and the entire payload is contained within a parent "data" object.
- Fixed a bug that caused launcher session failure on secrets that were expired on checkout but then disabled checkout via policy. Also, retroactively fixed this situation on secrets.
- Fixed a string truncation error. Expanded the user setting size to resolve issue for some customers with lots of columns for a grid.
- Fixed a timing problem where secret favorites might not initialize if the secret grid loads very quickly.
- Fixed an CSS issue where clicking the "Browse all folders" button caused folder names to overlap.
- Fixed an issue that had platform logout redirecting to a different tenant.
- Fixed an issue where "Web Launcher requires Incognito Mode" was not being respected when enabled. Descriptive text added on web launcher mapping for restricting input fields.
- Fixed an issue where a bulk action was applied to all secrets when select all is checked but a template or folder filter was applied.
- Fixed an issue where a default error page was presented when accessing certain URLs as opposed to a more technical error.
- Fixed an issue where a dependency fails to work when moved to another group or order due to the run condition. When a secret dependency is updated to the first sort order or to a new group, the run condition is

cleared. This addresses an issue where a secret dependency with a run condition would not run when it was the first secret dependency in its group.

- Fixed an issue where a Get Folders API call did not returns all descendants, breaking some customer integrations. To retrieve direct children only, use the new LimitToDirectDescendents parameter.
- Fixed an issue where an HSM could not be disabled.
- Fixed an issue where Azure domain accounts were unable to access Secret Server SSH Terminal with a public key. You can now log into Terminal with an Azure Active Directory account using SSH Key Integration. AAD logins to Terminal via password cannot be done.
- Fixed an issue where changing the time zone on the secret audit page did nothing and refreshing the page returned to the default time zone. When the server time zone is different, the time zone picker should show and the date column for audit should render in the selected time zone.
- Fixed an issue where connecting using an SSH key on another secret did not work with "SSH key only" secrets.
- Fixed an issue where data retention under PII removed monitored recordings or user audits related to monitored recordings. Data retention under database size management will still remove monitored recordings and related user audit records.
- Fixed an issue where duplicate user names were throwing an error. When logging in as a local user, we now ignore any Platform native users that may have the same login name, instead of erroring.
- Fixed an issue where event pipeline email notifications were not sent if the email task had an email template selected.
- Fixed an issue where exported computer scan logs were incomplete. Discovery logs now export more than 250 records.
- Fixed an issue where folders and sub-folders were missing secrets with UAM enabled. Left nav max folders default limit increased to 1,000. Setting dialog added to set the user preferred limit. Folder browser now loads 100 records at a time on scroll instead of just 30.
- Fixed an issue where GET /internals/secret-detail/{id}/launcher/{launchertypeid} threw an exception. We now show a friendly error message when launching a secret With Jumpbox Route with RDP that it is missing an SSH launcher
- Fixed an issue where OpenLDAP directory services group-search filter was not working.
- Fixed an issue where PowerShell dependency changer arguments were not being passed into the script.
- Fixed an issue where secret field data over a certain length may be rejected by the database upon replication.
- Fixed an issue where Secret PasswordComplianceCode was not updated after password field/PasswordReq change.
- Fixed an issue where secret template fields of type file no longer showed the drop down options when editing the field.
- Fixed an issue where session monitoring grid view showed the system and not the Secret Server. The secret session search date in the grid and card both now show in the selected time zone and the grid has the timezone picker when relevant.

- Fixed an issue where the folder list disappeared if UAM is enabled and when the "All Folders" toggle is selected in the sidebar. Folders in the tree will now be limited to only show 125 folders per tree. Once there are 125+ subfolders a "Browse all folders" option will appear in the folder tree. This link will take the user to a grid that only shows folders with a search. The grid has paging so it will load 30 folders at a time as the user scrolls. This will help support instances when users have thousands of subfolders. If there are more than 30 subfolders in a folder the secret grid will show a link to the new folder browser. This used to open a dialog to the folder tree which would also run into performance issues when users had over 1,000 subfolders.
- Fixed an issue where the folder tree disappeared when there were more than 1,000 folders accessed and UAM was enabled.
- Fixed an issue where the pipeline activity status stopped updating after the "Send to Email" task
- Fixed an issue where the Preserve Client SSH Process did not appear for process launchers
- Fixed an issue with heartbeat failures if a secret had checkout enabled.
- Fixed an issue with secret search would produce an excessively long URL that would sometimes throw an HTTP 404 error. The secret search API endpoint now accepts a filter param called ExtFieldsCombined, which is a comma delimited list of all extended fields to include in the results. This field is now used by the secret grid to help reduce the size of the URL when many secret fields are exposed for display to avoid the IIS 2k length restriction on GETs.
- Fixed an issue with SSH proxy "Tunnel RDP Connections" performance degradation (high CPU usage).
- Fixed an issue with the data retention page background color.
- Fixed discovery network view to ensure when searching you should be able to find all items under your current levels. However, when looking at a level you only see that level.
- Fixed issue with the password compliance report updating very slowly or not refreshing after either a template or direct PasswordRequirement password field change.
- Fixed issues related to RabbitMQ channel and queue growth and corruption-related issues due to connection interruption causing premature queue deletion.
- Fixed the default timestamp format for CEF.
- The "All Secrets" CSV download now correctly shows the folder name instead of the folder ID.
- The secret policy approvers "default only" option now displays correctly when updated.
- Updated the advanced session recording agent version label on the agent issues page to correctly state that it is the minimum required version, not the current version.
- Fixed an issue where a purge of inactive sessions longer than three minutes was occurring when the Sessions Monitoring page was displayed. It did not take into account the SSH proxy timeout. The page now obeys the timeouts.

Future and Recent Deprecations

Note: This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server Cloud Release Notes for March 31, 2023

Release Dates and Notes

Cloud: March 31, 2023 (including April 4, 2023 update)

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.6.0

Protocol Handler: 6.0.3.26

Enhancements

- Secret MFA: Block indirect API calls (Updated secret access logic for MFA).
- Updated disaster recovery data integrity checks to prevent replication when the replica is a higher version than the primary. In this scenario, the replica may have different schema and system data than the primary, causing replication issues.
- During disaster recovery replication, secret items from the source are combined with ones from the replica when they have matching SecretIDs and SecretFieldIds.
- "Save" buttons are no longer disabled when a form is invalid. Clicking the button will now show and trigger form validation messages.
- Color palette updated to improve accessibility and brand.
- Added option to duplicate a discovery scanner.
- Added the ability to sort in the new Discovery Scanner UI.
- Added credential validation message for the new Discovery scanner page.
- Launcher icons updated on secret general and inline secrets.
- Updated the new scripts UI to allow testing of remote scripts.
- Updated Delinea Platform opt-in flow to allow region selection and have explicit confirmation.
- Secret panel is now always open when on any secret section page. My Secrets page added to give a quick overview or dashboard of secrets, including favorites, recent, and search secrets and folders. My Secrets has a session secrets section which will show any secrets that have been opened during the web session. Favorites, recent, most used, and shared with me are now linked in the secret panel. Pinned folders now show a scoped version of My Secrets and only search within that pinned folder. New folder icon is always on top of the secret panel. A guide walkthrough will appear the first time a user goes to secrets. Folder URL path is now contained in /secrets/folder... Secret detail URL is now plural to be consistent and prevent reloading of the secret panel.
- A new checkbox has been added which enables requiring all users who log in through the Delinea Platform to have used Platform's multi-factor authentication when logging in.
- Unlimited Admin notification now appears in the header instead of floating as a transparent block.
- Unlimited admin page in configuration preview now has a link to open the unlimited admin audit.
- CreateUser.aspx now indicates that provisioning is in progress if accessed during the unified trial (platform) provisioning flow.

- When pages still using the legacy UI underpinnings, the Unlimited admin notice will show on .aspx pages when it is enabled.
- Added a Discovery import rule to the new network viewer.
- SDK Client Management pages have been converted to the new UI page format.
- The password changers list/grid has been updated to the new UI design.
- Converted list options ss-grid to thy-grid. Allows for resizing of columns.
- Discovery rules and dependencies grid can now be filtered by discovery source. Rule grid now also has discovery source available as a column.
- The secret detail page now includes a button to copy the current URL to the clipboard with rich text, including the secret ID and secret name. Plain text copy will just include the URL.
- Read-Only mode can now be enabled in Secret Server Cloud on the Disaster Recovery Configuration page.
- Local Admin column added to the new Discovery network view.
- Secret template name on the secret general tab is now a direct link to the template.

Bug Fixes

- Fixed a bug that caused launcher session failure on secrets that were expired on checkout but then disabled checkout via policy. Also, fixed this situation on secrets retroactively.
- Fixed issue so that when a PBA URL is configured through the UI, the host and certificate are now validated.
- Updated file-attachment selection process for data replication so that file attachments for secrets in the root folder are selected when the user has specified a folder block list.
- Fixed a problem with Azure AD / OpenIDC in April 1st monthly release prevented login.
- The report CSV download now encodes specific Turkish characters and has an updated mime type of text/csv.
- The sessions monitoring page used to purge inactive sessions longer than three minutes, disregarding the SSH proxy timeout. Now, it abides by the specified timeouts.
- The export secrets feature has been changed into an asynchronous job. The export process starts a job and continues polling until completion. Users should remain on the page until the job finishes to avoid timeout issues.
- An issue was resolved where a secret dependency with a run condition would not execute when it was the first secret dependency in its group or moved to a new group. Now, updating a secret dependency to the first sort order or a new group clears the run condition.
- The all secrets CSV download now correctly shows the folder name instead of the folder ID.
- Folder permission now correctly shows "None" in secret role drop down when in edit mode.
- Dates in the report export no longer include the "Z" for UTC when server time is used and ISO date format is selected, since the date/time is the server configured time and not necessarily UTC. This means the date is in ISO format and the server configured time, but does not include the offset. In some specific configurations when user format was selected, the time zone offset would be applied based on the actual server time zone and not the configured time zone.

- After changing field properties on a secret template, the UI cache is cleared to allow selectable columns in grids to be updated without requiring a browser refresh.
- Corrected a bug where converting or duplicating a secret with an assigned secret policy would cause launcher settings to apply multiple times, causing a UX constraint violation.
- An audit entry is now made for the user that enabled maintenance mode during an upgrade (on-premises only).
- Event subscription now publishes the event for when a user is enabled or disabled.
- Discovery logs will now export more than 250 records.
- Logging into Terminal with an Azure Active Directory account using SSH Key Integration is now possible. AAD logins to Terminal via password cannot be done.
- License server activation grid updated to resolve layout clipping issues.
- Secret dependency API variable name changed from id to secretDependencyId to help clarify which parameter is needed.
- Deleting folders will now also indicate that subfolders will be removed as well.
- Due to security reasons, we are removing the GET endpoint for /secretserversettings/export and replacing it with a POST endpoint where we can transmit the password securely. The contents of the payload are the same, except for the addition of "password" and "doubleLockPassword" fields, and containing the entire payload within a parent "data" object.
- Lookup folders (api/v1/folders/lookup) and search folders (api/v1/folders) will now return only direct children when searching by parent ID. They will no longer return grandchildren.
- Enabling heartbeat for the first time on a secret template will no longer subtract 1 minute from the heartbeat duration.
- The duration field on session monitoring now shows as a friendly time duration instead of just total seconds.
- "Automatic sudo or su privilege elevation" was fixed to work with Solaris OS.
- Connect using SSH key on another secret now works with SSH key only secrets.
- Addressed issues where secrets configured for checkout on the source could cause errors on the replica.
- Added documentation in a tooltip to point users to audit on proxy page (proxy audit).
- FOLDERPATH parameter now works with report schedules and running a report.
- The secret search API endpoint now accepts a filter param called extFieldsCombined, which is a comma delimited list of all extended fields to include in the results. This field is now used by the secret grid to help reduce the size of the URL when many secret fields are exposed for display to avoid the IIS URL length restriction on GETs of 2,048 characters.
- Inline row added to secret dependency log dialog to expand the row.
- Edit inbox rule condition dialog title now says "Edit condition" instead of "Add condition."
- Fix to allow heartbeats even if the secret has checkout enabled.
- Addressed an issue with disaster recovery replication where replicated custom launchers were not visible on their associated secrets.

- The report SQL editor no longer has options to download or configure columns on the report as it is not supported in that mode.
- Disaster recovery data replication errors caused by out of sync encryption keys are now automatically resolved properly.
- The "Most used secrets" grid on dashboard overview now downloads the folder path instead of the folder ID.
- The secret search API now returns the folder path on the secret. Secret grid download now includes folder path on all records accordingly.
- Fixed issues related to RabbitMQ channel and queue growth. Channel and queue growth should no longer be experienced.
- Resolved an issue with disaster recovery folder synchronization selection. Personal folders can now be selected for either allow or block lists.
- Fixed an issue with folder name collisions during disaster recovery data replication sync.
- Fixed older character sets that failed to replicate when running disaster recovery.
- Handled issue when replicating data for disaster recovery where pre-existing users on the replica that do not exist on the source could lose their All Vault Users group membership.
- Descriptive text added on web launcher mapping for restricting input fields.
- Fixed an issue with the secret log length validation in the UI.
- Fixed issue where secret field data over a certain length may be rejected by the database upon replication.
- Corrected an issue where when created as part of the unified trial flow for Platform, the first user created did not have admin access to Secret Server Cloud.
- Folders in the tree will now be limited to only show 125 folders per tree. Once there are 125+ subfolders, a "Browse all folders" option will appear in the folder tree. This link will take the user to a grid that only shows folders with a search. The grid has paging, so it will load 30 folders at a time as the user scrolls. This will help support instances when users have thousands of subfolders. If there are more than 30 subfolders in a folder, the secret grid will show a link to the new folder browser. This used to open a dialog to the folder tree which would also run into performance issues when users had over 1,000 subfolders.
- Fixed issues for password requirement character set data replication in the DR feature.
- Corrected the default timestamp format for CEF.
- Expanded the user setting size to resolve issue for some customers with lots of columns for a grid.
- Left navigation maximum folders default limit increased to 1,000. Setting dialog added to set the user preferred limit, folder browser now loads 100 records at a time on scroll instead of just 30.

Future and Recent Deprecations

Note: This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server Cloud Release Notes for February 11, 2023

Release Dates and Notes

Cloud: February 11, 2023

Component Versions

Distributed Engine and Advanced Session-Recording Agent: 8.4.3.0

Protocol Handler: 6.0.3.26

New Features

User Interface Streamlining: Classic UI Removed

The classic UI is no longer available as an option, and can no longer be enabled. This followed notifications of phased deprecations in prior releases. Several improvements have been made to the UI based on feedback from customers regarding this change.

Checkout Extension Maximum Limit

We created a global configuration setting that allows administrators to set a maximum secret-checkout extension interval. This provides additional admin control by specifying granular limitations to users extending a checked out secret. The time limitation begins at the point of checkout extension and extension time defaults to the set checkout time

Disaster Recovery Enhancements

"Replicated User Status on Disaster Recovery Source" configuration can now be set to:

- Mirror Source, which is the existing behavior
- Disabled by default

New Synchronization Items:

- Character sets
- AD users and groups

Discovery User Experience Improvements

We updated the discovery user experience to reflect the style and design of the application. The legacy pages are still available; however; the new interface items are ready for use, and we welcome feedback on these items. The legacy pages can be accessed by browsing to the relevant new interface and clicking the "View Legacy Page" button. The improvements are:

- Network view is available as a tab on the main Discovery Sources page.
- Network view displays the same results as the legacy page, in a single filterable grid, as opposed to individual tabs for different types of scanners. The new page has the functionality of the legacy page but in a more-responsive and updated design.
- There is a new filter menu, allowing extensive filtering options of this data.
- Each item in the network view list links to a details page allowing review of the discovered data, as opposed to being viewed inline.
- The grid data is exportable as a CSV, similar to other grids.

- Scanner, scan template, command set and search filter configuration are available from the Discovery Configuration Options > Scanner Definition button on the Discovery Configuration page.
- Source scanner configuration is available in the Discovery Source Configuration page as a tab.
- We extensively redesigned the scanner configuration UI to make this experience more intuitive, and scanners are now displayed in a workflow view.

Generated and Created Password Improvements

Password Complexity Indicator

There is a new visual indicator in the password complexity rules that provides the user with a better understanding of the strength of their password. The combined score considers both entropy score (brute force defense) and character limitations (social engineering defense). In the case that the score is deemed too low, the UI provides recommendations to the user on how to increase password strength

New Password Rules

We introduced character rules to password complexity selection to enhance the strength of generated and created passwords, if enabled. The new rules provide flexibility in the granularity of the rules. Each selection impacts both entropy and overall strength score. The rules include minimum characters from:

- Lowercase letters
- Symbols
- Numerals
- Uppercase letters

Opt-In Engine Upgrades

Distributed engine upgrades are no longer mandatory for every release. We added a new setting to the Distributed Engine Configuration page to set the minimum required engine version. Modifying this will trigger an automatic update for any engine below this version.

In the action menu for an engine on the Sites page, a manual upgrade can be triggered for individual engines below the latest version, which prompts the engine to update when it next calls in.

When changes are made needing an upgrade, the minimum required version is updated during the update process, and all engines update immediately.

"Run Scripts" Role Permission

We created a new "Run Scripts" role permissions to separate privileges in script management. Holders of the "View Scripts" role permission cannot execute test runs of scripts, and the new role permission must be assigned to perform this task.

Administer Scripts remains unchanged and allows view, edit, and run permissions.

Syslog Timestamps

There is a new setting in Syslog configuration allowing the selection of timestamp formatting. The standard for Syslog indicates that ISO timestamps should be used; however, some consumers use the legacy format. There is now a selection between Syslog and ISO format. Syslog will be the default for upgrades to allow current configurations to retain their behavior, and ISO format is the default in new instances.

Enhancements

- Added a configuration option to disable the SMB heartbeat fallback check.
- Added a secret policy setting to control "Change Password Upon Check-In" behavior. Previously, this was automatically enabled if "Require Check Out" was enabled.
- Added additional debugging output for SSH proxy when using the "ALL" logging level.
- Added audits for emailing and downloading reports.
- Added endpoints for Update Password Type Auth, Get Password Type Auth, and Create Password Type Auth. These allow you to create and update records for the command arguments on RPC command set up.
- Added the configuration setting "Allow Files without Extension" to the configuration preview.
- Added the internal site connector configuration to the configuration preview.
- Added the User parameter to the IBM iSeries Mainframe connection for launching, password changing, and heartbeat.
- Bulk edit share now has a "None" permission which allows removing permissions.
- Changed IIS web.config configuration to disallow access to the file uploads folder.
- Enabled more connection classes to use read-only mode.
- Enhanced secret export logging.
- Improved performance of dependency matching within discovery.
- Improved performance of the role assignment page and added a user panel on the same page.
- Improved performance of the secret to computer matching operation that runs as part of discovery.
- In the data replication summary log now lists in alphabetical order, success and version number will appear before the list of items and any errors are appended at the end.
- Optimized application caching.
- Updated the new UI to allow newly generated SSH keys to have a blank passphrase, which matches legacy UI functionality.

- ### Bug Fixes

- Fixed a memory leak in SSH proxy.
- Fixed a SAML audit error.
- Fixed an error that occurred when multiple identical domains were created.

- Fixed an error where the new SSH proxy custom SSH cipher suite settings were not picked up by distributed engines.
- Fixed an issue in the heartbeat status by day report that would cause the same secret to be counted twice on days where the secret transitions between heartbeat failure and success.
- Fixed an issue where "Days Until Expiration" value on the secrets grid would show a large negative number if expiration is forced. This now displays "Expiration forced."
- Fixed an issue where "requires approval type" could not be set by policy.
- Fixed an issue where a user with edit permissions could not rename a folder.
- Fixed an issue where an "invalid SQL error" was incorrectly displayed when a report timed out.
- Fixed an issue where an error was displayed in an edit field dialog box.
- Fixed an issue where an inbox notification was not clearing in Secret Server Cloud.
- Fixed an issue where completed master encryption key rotation would not show as such.
- Fixed an issue where converting a secret in a folder with a launcher settings policy threw an error.
- Fixed an issue where deleted Active Directory groups were not correctly marked as inactive when synchronized.
- Fixed an issue where disaster recovery would log many password requirement errors.
- Fixed an issue where failing Syslog/SIEM messages did not respect updated Syslog Server configuration.
- Fixed an issue where file contents during SFTP/SCP file transfers were included in the session keystroke recordings.
- Fixed an issue where installation on specific dates on servers with a dd/mm/YYYY localization configuration would prevent some configuration settings from being read.
- Fixed an issue where Local site could not be configured to use Custom SSH Cipher Suite settings when set to process on the Web Site.
- Fixed an issue where manual backup did not work in maintenance mode.
- Fixed an issue where master encryption key rotation would fail due to discovery import rules running at the same time.
- Fixed an issue where missing file attachments caused DR replications to fail.
- Fixed an issue where monitoring and termination of live sessions was not displayed in the UI. This now takes the user to the regular session playback page, which displays the live session.
- Fixed an issue where PowerShell-based dependency changers would not correctly pass arguments.
- Fixed an issue where PuTTY would close immediately following a session error. This was due to a default setting change in PuTTY, which is now explicitly set to remain open on installation or update of the protocol handler. This requires a protocol handler update.
- Fixed an issue where reports generate an application error if users navigated away from the report while it was loading.
- Fixed an issue where scrolling the secrets grid view would deselect items.

- Fixed an issue where Secret Server did not correctly react when two templates have the same field if one had spaces that the other did not.
- Fixed an issue where secrets would not open when users have folder view and secret list permissions. The secret audit within that folder should be accessible.
- Fixed an issue where session recording would sometimes show a 500 error, even though the client would retry. Replaced this with a HTTP 429 response, explicitly informing the client to retry.
- Fixed an issue where session recordings could not be saved to a UNC file path due to missing permissions on the root of the path.
- Fixed an issue where sorting by folder path in the secret grid view would return an error.
- Fixed an issue where SSH dependencies would not process on distributed engines.
- Fixed an issue where SSH Proxy would not allow a launcher to connect in maintenance mode. This is now possible in non-recorded sessions—recording is not possible in maintenance mode.
- Fixed an issue where the advanced session recording agent would attempt to make many reconnections in a short time span.
- Fixed an issue where the 64-bit protocol handler would not function when "Enable Protocol Handler Auto-Update" was enabled.
- Fixed an issue where the folder picker would not populate when adding folders in event pipeline policies while in unlimited admin mode.
- Fixed an issue where the IBM iSeries password changer was not properly adding the model value to the connection string.
- Fixed an issue where the light/dark mode toggle displayed "Enable Dark Mode" even though the UI was already in dark mode. This was due to a dark mode browser preference and no explicit user preference having been set.
- Fixed an issue where the SAML AuthnRequest was sending a blank RequestedAuthnContext when Authentication Context was set in the Identity Provider Configuration.
- Fixed an issue where the Secret Server website would not load if the internal site connector is unavailable at startup.
- Fixed an issue where the terminate option on the session playback page was missing.
- Fixed an issue where the test buttons would not function for the Oracle Account Ver. 2 password changer.
- Fixed an issue where the UI session monitoring search required additional permissions to load.
- Fixed an issue where the unlimited administrator watermark could block interaction with some page elements.
- Fixed an issue where the wrong error message would be shown when trying to apply an invalid data source key under data replication.
- Fixed an issue where user permissions on replica instances were removed erroneously when data replication ran.
- Fixed an issue with check in when the "Check In Secret on Launcher Close" and "Close Launcher on Check In Secret" settings were both false.

- Fixed an issue with DR replication where some operations would give an error "The incoming request has too many parameters."
- Fixed an issue with excessive CPU usage for RDPWin.exe. Protocol handler and session connector no longer track or record processes using WMI. Now they use native Windows calls, reducing CPU usage of the Windows WMI Provider. The exception is when "Run as secret credentials" is used—it still uses the WMI process tracking.
- Fixed an issue with replicating domain users. This now correctly links the user with the replicated domain.
- Fixed an issue with slow loading sessions failing to load when using session connector. This required an update to the latest protocol handler (RDS) on the session connector server.
- Fixed an issue with the group filter in event pipelines to ensure precise name matching is correctly used.
- Fixed an issue with the SearchSecretsByFieldValue SOAP API function that caused it to return a 500 error.
- Fixed bug where email filters click through approval links.
- Fixed an issue where OIDC platform connection failed for previously imported users after a domain change.
- Mitigated the possibility of an error in SSH Proxy command processing.
- Removed parameters from ASRA installer to accommodate long secret URLs.
- Resolved an issue with RADIUS challenges in Secret Server Cloud.
- Session recordings which are invalid due to no data are now recorded as an error to prevent failure upon playback.
- Updated discovery SSH scanners to handle messages coming back without the stdout marker.
- Updated logging around Azure AD Sync to make it clearer when the sync stops due to configured groups missing in Azure AD.

Future and Recent Deprecations

Note: This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server Cloud Release Notes for December 3, 2022

Component Versions

Distributed Engine: 8.3.1.0

Protocol Handler: 6.0.3.23

New Features

None

Enhancements

- Improved custom date functionality when running data replication. Users can now access it through a dialog that pops up when clicking "Run Data Replication."
- Added the User parameter to the IBM iSeries Mainframe connection for launching, password changing, and heartbeat.

- Improved performance of discovery dependency matching.
- Added "Run Scripts" role permissions to ensure users with the "View Scripts" role permission cannot run test scripts.
- Improved the secret grid-view filter and column settings.
- Added extra logging for CSV export failures.

Bugs

- Fixed an issue where service dependency changers would fail due to Windows using a different naming format than the secret. Added error handling for several possible cases.
- Fixed an issue where session recording would sometimes show a 500 error, even though the client would retry. Replaced this with a HTTP 429 response, explicitly informing the client to retry.
- Fixed a memory leak in SSH proxy.
- Fixed an issue with slow-loading sessions failing to load when using session connector.
- Fixed an issue where launchers were not working in maintenance mode for proxied connections.
- Fixed an issue where PuTTY would close immediately following a session error. This was due to a default setting change in PuTTY, that is now explicitly set to remain open on installation or update of the protocol handler.
- Fixed an issue where the checkout timer expired and caused an infinite redirect loop resulting in the secret page blinking after check-in time out and the secret still in use for RDP.
- Fixed an issue where "Days Until Expiration" value on the secrets grid would show a large negative number if expiration is forced. This now displays "Expiration forced."
- Fixed an issue where converting a secret in a folder with a launcher settings policy threw an error.
- Fixed an issue where the folder picker would not populate when adding folders in event pipeline policies while in unlimited admin mode. Other areas with the same dialog should be checked too.
- Fixed an issue where master encryption key rotation would fail due to discovery import rules running at the same time.
- Fixed an issue where the test buttons would not function for the Oracle Account Ver. 2 password changer.
- Fixed an issue where the secrets.csv file was not downloaded.
- Fixed an issue with the group filter in event pipeline to ensure precise name matching is correctly used.
- Fixed an issue with an activation banner saying activation was required but no entries where returned from the activation site.
- Fixed an issue where spatial pattern validation password requirements with large passwords took a long time to generate or threw a 500 error.
- Fixed an issue where PowerShell-based dependency changers would not correctly pass arguments.
- Fixed an issue where file contents during SFTP/SCP file transfers were included in the session keystroke recordings.
- Fixed an issue with disaster recovery synchronization not replicating deleted launcher mappings.

- Fixed an issue where disaster recovery would log many password requirements errors.
- Fixed an issue where the light/dark mode toggle displayed "Enable Dark Mode" even though the UI was already in dark mode. This was due to a dark mode browser preference and no explicit user preference having been set.
- In the summary log for Data Replication, items replicated will now be listed in alphabetical order, success and version number will appear before the list of items and any errors will be appended at the end.
- Added the internal site connector configuration to the configuration preview.
- Optimized application caching.
- Fixed an issue where removing an RPC Schedule from a secret policy did not update a secret assigned to the policy.
- Fixed a SAML audit error.
- Fixed an issue with check in when the "Check in Secret on Launcher Close" and "Close Launcher on Check In Secret" settings were both false.
- Fixed an issue where sorting by folder path in the secret grid view would return an error.
- Fixed an issue where monitoring and termination of live sessions was not displayed in the UI. This now takes the user to the regular session playback page, which displays the live session.
- Fixed an issue where deleted Active Directory groups were not reflected as disabled during Active Directory synchronization. This now correctly registers as a disabled group and is disabled if configuration is set to "Mirror the AD Status."
- Fixed an issue where FileZilla failed to fallback if SFTP subsystem is not enabled on target.
- Fixed an issue where the UI session monitoring search required additional permissions to load.
- Fixed an issue where installation on specific dates on servers with a dd/mm/YYYY localization configuration would prevent some configuration settings from being read.
- Fixed an issue with RADIUS challenges in cloud.
- Fixed an issue where the terminate option on the session playback page was missing.
- Updated the IBM launcher name.
- Fixed a policy-activation validation issue for the "Secret Has RPC Enabled" filter.
- Fixed an issue where reports generate an application error if users navigated away from the report while it was loading.
- Fixed an error where the new SSH proxy custom SSH cipher suite settings were not picked up by distributed engines.
- Fixed an issue where missing file attachments caused DR replications to fail.
- Added a banner displaying the Classic UI deprecation planned for December 10, 2022.
- Fixed an issue where secret items were not hashed for indexing if updated from an empty value.
- Fixed an issue where an "invalid SQL error" was incorrectly displayed when a report timed out.
- Added a row button in secret view to indicate that rows can be expanded for preview.

Future and Recent Deprecations

Note: This section describes planned future deprecation of feature or platform support in Secret Server.

Secret Server Cloud Supplemental Release Notes

October 24, 2020

The following are changes to Secret Server Cloud that supplement the Secret Server on-premises release 10.9.000002.

Security

Added an administrator configuration option for Web Password Filler to require a fully qualified domain match.

Enhancements

- Scheduler role logging improved when a service provider disables an instance of Secret Server.
- Failure prevention processes for mitigating service outages improved.
- Delivery and loading of UI frameworks improved.
- Discovery dependency scanning performance improved.
- Discovery Active Directory OUs scanning via the engine worker improved.
- Error logging improved when a user enters an invalid AD credential.
- Secret Server Cloud platform stability improved.
- Secret names in reports are now links to the corresponding secret (also appears in 10.9.000002 on-premises release).

Bug Fixes

Fixes applying only to Secret Server Cloud:

- Fixed an intermittent issue with the bulk exportation of event logs.
- Fixed an issue where a secret's audit log accumulated an excessive number of encryption and decryption event messages.
- Fixed additional infrastructure bugs affecting the overall stability of Secret Server Cloud.

Fixes in common with the 10.9.000002 on-premises release:

- Fixed discovery rules to correctly handle OUs with bracketed names.
- Fixed issue so that on log off from Secret Server no longer sends the `Clear-Site-Data` header, which could log users out of unrelated Web applications.
- Fixed issue where SSH connections via SSH proxy closed incorrectly.
- Fixed an SSH proxy connection timeout when connecting via a distributed engine.

Secret Server Cloud: Release Notes 2019-12-21

Upgrade Notes

- This release launches a new web password filler. To update your web password filler extension, go to the extension download site for your browser and platform.
- Users will be directed to the dashboard Overview tab for their first login after upgrading.
- New Secret Server Cloud (SSC) installs now have stricter default local password policies. The policies remain customer configurable. Go to **Admin > Configuration > Local User Passwords**.

New Features

Data Retention

Secret Server now allows administrators to permanently delete audit records for tables that either contain Personal Identifiable Information (PII) or tables that can grow large in enterprise environments. To configure these settings admins need to add the permission "Administer Data Retention" to the user's role and then the user can navigate to **Admin > Data Retention**. Only users with the "Unlimited Administrator" permission can assign this new permission.

Time-Based One-Time Passwords (TOTP)

Added a new feature where Secret Server can now generate time-based one-time passwords (TOTP) for web secrets. This allows users to implement TOTP on shared secrets. Configuring secrets for TOTP begins at the secret template level.

Truncated Log Data

Added the ability to truncate table logs for several types of data that log to the "Status Message" table. These messages can contribute to excessive log data and slow performance. The option to truncate each message type is called "Days to Keep Operational Logs" and is under the "Advanced" sections on the following list of configuration pages. Minimum message retention time is one day and the default is 30 days. The logs include:

- AdminDiscovery.aspx (**Admin > Discovery**)
- AdminSearchIndexer.aspx (**Admin > Search Indexer**)
- ConfigurationActiveDirectory.aspx (**Admin > Active Directory**)
- ConfigurationPasswordChanging.aspx (**Admin > Remote Password Changing**)
- ConfigurationSshProxy.aspx (**Admin > Proxying**)
- ConnectWiseConfiguration.aspx (**Admin > Folder Sync**) Setting only available when using the "Database" Folder Synchronization Method on this page.

Technical details: A background task was added that scans the status message table every 12 hours and checks the status messages against configured values for how long they should be retained. These configured values were added to applicable UI pages.

Web Browser Extensions

The Web browser extensions for Secret Server were rebuilt with a new look and feel and now have additional browser and site support. These new plugins are available for:

- Google Chrome
- Mozilla Firefox

These features from the old browser plugins have been improved to allow more flexibility:

- Create secrets
- Select secret template
- Generate complex password

Users can now authenticate to Secret Server directly from the Web plugin, including support for 2FA options, such as DUO. Log in via Secret Server is also available for users with single sign-on, SAML, or other multi-factor authentication mechanisms. Web plugins automatically identify manual entry of new credentials in a Web page and offer to save the credentials as a secret. There is also improved support for sites that use multi-page login mechanisms.

Enhancements

Advanced Session Recording

Added a new setting to disable keystroke data from advanced session recording metadata. The new setting is called "Default Keystroke Recording Configuration" and can be configured under **Admin > Configuration > Session Recording > Configure Advanced Session Recording.** Click **Collection name** to edit individual collection settings or agent settings. By default, advanced session recording keystrokes are enabled.

Database SQL Indexes

Added new SQL indexes for the following areas:

- Column LauncherSessionGuid on the Launcher Session Video (tbLauncherSessionVideoSegment)
- Event Queue Monitor (tbEventQueue)
- Expired Secret Monitor (tbSecretDependency table)
- Folders (tbFolder, tbFolderGroupPermissions)
- General Navigation (tbUserSession)
- Launcher Activities (tbSecretSession)
- Log In (tbUser)
- Node Activation Check (tbNodeLicenseActivation)
- Secret Log table (tbSecretLog)
- System Reports (tbAuditUser, tbAuditSecret)

Discovery

Added messaging for when computer or dependency scans do not run due to having no scanners configured for a discovery source.

Distributed Engine Offline Status

Updated the definition of distributed engines' offline status to be the configured heartbeat interval times three. For instance, if your heartbeat interval is configured at 5 minutes, the engine will report offline if Secret Server and the engine do not successfully communicate within a 15-minute time period. Engine online and offline states were also added to subscription actions to allow notification to admins when engine states change.

Licensing

A second distributed engine is now available, by default, for the local site.

New User Interface

- Redesigned the Admin landing space. Click **Admin > "See All"** to explore the new layout.
- Redesigned DoubleLock.
- Added new "Recent Activity" section to the Home dashboard page to display recent activity at a glance.
- Updated the Security Hardening tab in the Reports page.
- Updated the IP Address Management pages under Admin.
- Added custom logos. Added custom "full-sized" and "collapsed" logos for the new UI in **Admin > Configuration** under in the **User Interface** section.
- Added dark mode theme option in the new UI. To change theme mode preferences, go to **Account Settings > Color Mode**. Options include Light Mode, Dark Mode, or Default (mode will update based on user's OS color mode settings).
- Added a new setting to configure the inactivity time before the new UI goes into dark screen "sleep mode." To configure go to **Admin > Configuration > User Experience > UI Inactivity Timeout**.
- Converted the Groups page to the new UI.
- Updated error messaging in the new UI to display folder synchronization and deletion errors.
- Updated the date picker to allow for future start dates and time selection without first adjusting the end date when requesting secret access. End dates are automatically adjusted to align with the start date +1 hour.
- Updated grid downloads in the new UI to download according to new options. User options now include choices to download all data or specific rows of data, and specify date format. You can also choose time zone options of UTC, server time zones, or the local browser time zones.

  Note: for downloaded reports users' time zone options are limited to UTC or the server time zone.
- Updated behavior of new UI so that clicking the "Select All" check box at the top of a secret grid selects all rows. Previously the check box selected only the items currently loaded on the page.
- Added the "View Audit" button to the reports page of new UI.
- Added the "Upgrade Available" banner to display in the new UI.

- Added the ability to drag-and-drop child folders into the root folder. Folders will automatically re-order alphabetically in the left navigation pane.

  Note: This action is only allowed if users have the "Create root folder" permission and own folders that they are attempting to move.
- Added folders to the "Shared With Me" page.
- Added new inbox notifications including "getting started" notifications for new installs and administrator alerts when an instance is close to hitting licensing limits.
- Added the ability to mark Inbox notifications as read or unread for most notification types.
- Added the ability to browse by folder name using the URL format `[SecretServerURL]/SecretServer/app/#/lookup?folderPath=[FolderName]`. If multiple folders exist with the same name, this URL search schema only directs users to the first folder listed within the left navigation pane.
- Updated Favorite star icons to remain in column view when the Name column is resized.
- Expanded file-size allowance on file uploads. File uploads can now be up to 10 MB.
- Grid results updated to auto-load 30 results instead of 15.

Remote Password Changing upon Regex-Defined Error

Added a new regex setting to automatically retry a remote password change (RPC) with a regenerated password if the original RPC failed due to a specific type of error.

Go to **Admin > Remote Password Changing**, click **Advanced** under the **Configure Password Changers** section. The new setting is **Attempt Password Change with new password when error contains (regex)**. Edit it to provide the regex failure code that will trigger the automatic next password RPC.

Reports

- Updated several reports to no longer show deleted secrets.
- A new out-of-the-box report called "Secret Templates without an expiration field" was added to display any secret templates that have a password field but do not have an expiration field set.

Secret Template Import and Export

Updated secret template settings for importation and exportation to include:

- Is Required?
- Edit Requires
- Hide on View
- Secret template icon
- Keep Secret Name History
- Validate Password Requirements on Create/Edit
- Field Slug Name

- Type Description
- One Time Password settings

The secret template settings that do **not** transfer include:

- Launcher settings
- Password changing settings
- Session recording enabled
- Associated secrets

SSH Proxy

- Updated "connect as" to accept key-based SSH authentication without also requiring a manual password.
- For SSH proxy sessions, added the option set:
  - Only record keystrokes
  - Only record video for sessions.

  By default new installs will only record keystrokes on SSH proxy sessions to preserve disk space. To configure this setting go to **Admin > Configuration > Session Recording tab > Secret Server Proxy Session Recording**. Edit the **SSH Proxy Session Recording Options** dropdown list. The options include:
  - Record keystrokes and video
  - Record keystrokes only
  - Record video only
  - Do not record

Terminal

- Added terminal instructions for how to view SSH proxy credentials in the new UI under **Secret Options > Show SSH Terminal Details**.
- Removed restrictions from the allowed number of concurrent logins for SSH terminal. Previously, terminal logins were tied to the "Maximum concurrent logins per user" setting that establishes this number for UI-based users.
- Added Unicode support for SSH command menu items (names and descriptions).
- Added "clear" command to terminal.

Unique Field Slug IDs

Added a new "Unique Field Slug" ID column for secret templates to allow users to create secrets with duplicate field names without compromising the ability to target each field name with a unique identifier for API calls.

User Variables for Scripting

Added the following user-based script variables to be used in API calls as arguments:

- `$SECRETSERVERUSERID`
- `$SECRETSERVERUSERNAME`
- `$SECRETSERVERDISPLAYNAME`
- `$SECRETSERVEREMAILADDRESS`

This allows, for example, that when a specific user runs a check-out hook, they can pass a user email, ID, username, or display name as a parameter into the script to use a check-out hooks and related AD functionality in Secret Server through the API.

Verbose Logging

Added Verbose Logging for:

- AWS password changers
- AWS discovery scanner
- ComPlus dependency scanner
- PowerShell discovery scanner
- Flat file discovery scanner
- ODBC discovery scanner
- SSH discovery scanner
- ESX discovery scanner

API and Scripting

API General

Added a setting that allows users with view permission on a secret to get the secret's "autoChangeNextPassword" field in the API. This setting is enabled under **Admin > Configuration > Permission Options**. Set **Allow View User To Retrieve Auto-Change Next Password** to **Yes**.

New API Calls

- Get one time password code and seconds: `GET /one-time-password-code/{id}`
- Search secrets by URL: `POST /secret-extensions/search-by-url`
- Get AutoFill values for URL by secret ID: `POST /secret-extensions/autofill-values`
- Update secret field: `PUT /secrets/{id}/fields/{slug}`
- Update secret: `PUT /secrets/{id}/restricted`
- Get SSH Terminal details: `POST /secrets/sshterminal`
- Get extended regex values by secret: `GET /extended-fields/regex/{secretId}`

Removed API Calls

- Search app clients: `GET /app-clients`
- Update secret: `PUT /secrets/{id}`
- Update secret field: `PUT /secrets/{id}/restricted/fields/{slug}`

Integrations

- Added support for Open ID Connect to integrate with Secret Server authentication.
- Added Connection Manager as a tools option in the grid menu and under the Admin space in the new UI.

Performance Improvements

- Significantly improved shutdown and restart times for engines connected to SSC.
- Added server-side paging to reports in the new UI to address performance issues when attempting to load reports with large numbers of records.
- The new UI will no longer load the subfolders If a parent folder has more than 30 subfolders within it on the grid page. Instead, a folder picker will display above the folder's secrets that will allow users to select a specific subfolder.
- Applied enhanced SQL querying logic on the groups pages so that environments with large groups no longer experience page timeouts when processing group data.
- Improved the shutdown performance in distributed engine.
- Removed the welcome widget from the dashboard on the classic UI due to page load issues in large environments.
- Enhanced SQL query for the unlimited admin report to improve performance for large environments.
- Added a new "use database paging" setting for the custom reports page. Database paging allows the database to load large reports more quickly. We recommend database paging if the query is expected to pull large amounts of data for the report. Implementing database paging may not work if the SQL query uses some keywords, including TOP, OPTION, UNION, WITH, or aliases containing the word FROM.

  Example queries:

  - Works using database paging: `SELECT * FROM tbSecret WHERE NAME LIKE 'Test%'`
  - Does not work using database paging: `SELECT TOP 10 * FROM tbSecret WHERE SecretName LIKE 'Test%'`

Security

- Updated PuTTY to version 0.73. Updated version addresses several PuTTY vulnerabilities, including one critical and two high severity items. CVE-2019-17067, CVE-2019-17068, CVE-2019-17069
- Addressed a vulnerability with the SDK client account handler.
- Fixed a permissions issue in the new UI where password requirements did not obey the "administer custom password requirements" permission.

- Added audits and event subscriptions for viewing passphrases and SSH keys.
- Addressed a Remote Code Execution (RCE) vulnerability that allowed parameter changes for an action without validating user permissions.
- Resolved an issue for SSH scripts and SSH remote password changers where sensitive information was being written to log files:
  - SSH remote password changers will now only log the comment for each command as it runs.
  - SSH scripts will only log that they ran because they have no comment for each command.

  Note: If you manually test an SSH script or password changer, the full output will still be shown for debugging purposes, because you just entered the credentials yourself.
- Resolved a URL redirection vulnerability.
- Added configurable parameter quoting for custom launchers.
- Resolved three cross-site scripting (XSS) vulnerabilities.
- Fixed an XML external entity (XXE) injection vulnerability.
- Removed user information that was returned in an API call.
- Added auditing for changes made to the session recording configuration page on the **Admin > Configuration > Session Recording** tab.
- Added auditing for test script actions in the **Custom Command Edits** section in the **Admin > Scripts** pages.
- Added auditing to the **Admin > Configuration > Ticket System** tab. Audits are logged under **Admin > Configuration > General tab > View Audit.**
- Updated missing secure cookie attributes when "Force HTTPS" is enabled.
- Added SHA1 and SHA256 hashes for protocol handler.
- All Delinea DLLs and EXEs are now signed with the Delinea Software certificate, including distributed engine and advanced session recording agent.

Bug Fixes

- Fixed an issue where creating folders through the API failed to set a secret policy.
- Fixed a memory leak issue where leaving Mac launcher sessions open for extended periods of time consumed increasing amounts of memory on the machine hosting the session. This issue was incorrectly believed to be fixed in the Secret Server version 10.7.000002 release.
- Fixed an issue where Unix secrets were not reported in the "Password Last Changed" report because the Unix Account template did not have a password expiration field by default. Unix password fields are now set to expire at 30 days by default.
- Fixed an issue where pressing Enter with the cursor in the Search bar on the Discovery Network View page would open the create rule dialog.
- Fixed an issue where new users were not adequately loading in the dropdown option for subscribers in the "discovery rule alerts" setting if users increased from under 40 to over 40 users.
- Added clear error and validation in entering credentials for a discovery scanner.

- Fixed an issue where localized language customization did not apply to some product pages due to default cache keys or inconsistent HtmlHelper methods implemented on those pages.
- Fixed a bug where some pages in the old UI did not follow customized headers from CSS stylesheets.
- Fixed an issue where extended search terms were not applied for URLs. Updated BookmarkletSecretSearcher so that it will not do extended hashes on URLs that might result in many erroneous matches. For instance, `facebook.com` would match `face.org`.
- Fixed issue where non-AD discovery sources (ESX, PowerShell) would not match dependencies with domain to an existing AD Domain. If the dependency scan item has a field called "domain," it will attempt to map to an existing domain.
- Fixed a bug where the REST API "daysUntilExpiration" field returned a blank value when calling for a secret summary.
- Added a query to resolve sort ordering issues for dependency numbering.
- Fixed an issue in the new UI where deleted secrets remained on display on the favorites widget in the home tabs.
- Fixed a "bad token" error on login for mobile apps (Windows Desktop, iOS, or Android) for local users.
- Fixed an issue where SSH keystroke data was not searchable from the session playback page due to proxy session data not being correctly hashed in the database.
- Fixed a ticketing system bug where the option to require users to either provide a ticket number or a comment for requesting access incorrectly required both a comment and a ticket number to access the secret. This issue existed when requesting access to the secret through the UI, workflows, or terminal.
- Fixed a bug where hashed terms were intermittently slow to return search results in the new UI.
- Fixed a rendering issue for the group edit page when using the IE browser.
- Fixed a localization issue in the SSH command menus where setting non-English as the global or user preference threw an exception when trying to save a secret policy.
- Fixed an issue where an "object disposed" exception was thrown when navigating away from the new UI soon after application pools were recycled. This occurred because of an incorrect re-use of a service that is for processing the first Web request to the application.
- Fixed a bug in the new UI where personalized preferences for launcher settings on a secret were not allowing users with view access to save.
- Fixed a bug where custom columns on grids in the new UI occasionally tried to display the same column twice and threw an error.
- Fixed an issue where a syslog header was missing the hostname when logging from an engine.
- Fixed an issue where Duo authentication was not checking for valid fallback device options when the configured "default device" failed to authenticate.
- Updated the "find new dependencies" feature to be available to users with edit permissions on the secret. Previously the new UI required users to have owner permissions.
- Resolved a timing issue where secrets with scheduled password changes enabled would get erroneous heartbeat fails due to remote password change (RPC) expiration and heartbeat occurring at the same time. Now

heartbeats are skipped for an interval of five minutes before and after a scheduled password change to allow password change completion before heartbeat is attempted.

Secret Server Cloud: Release Notes 2019-09-21

Upgrade Notes

Secret Server Cloud users may need to perform a **hard refresh to clear local browser cache files** for certain 10.7 upgrade changes to take effect in the new UI.

New Features

AWS Discovery

Added discovery for AWS accounts.

SSH Terminal

Added an SSH terminal to Secret Server, allowing users to connect to Secret Server via SSH to search for secrets, access secret data, and initiate proxied SSH connections.

Integrations

Updated ConnectWise API calls to pass in the ClientID object due to version updates by ConnectWise that requires a ClientID for all API calls. Updates released by ConnectWise in Aug 2019.

Enhancements

Security

- Secret Server 10.7 uses jQuery 3.2.1, which is listed as vulnerable to the jQuery CVE-2019-11358. Updated Secret Server to resolve this jQuery vulnerability.
- Updated PuTTY version (0.71) in the Secret Server protocol handler to support the elliptic curve cipher for handling keys during the SSH connection process.
- Addressed a security issue where reusing Active Directory usernames could expose secret data.
- Resolved an issue affecting FIDO2 authentication. Security Issue Discovered by : Vladimir Skuratovich.
- Two cross-site scripting vulnerabilities were fixed in Secret Server.
- Removed a server-side request forgery issue in the legacy Web launcher.
- Addressed a security issue that could result in folder name disclosure.
- Addressed a security issue with dual control where an approver could approve their own access.
- Resolved a URL Redirection issue.

New User Interface

Drag and Drop Folders

Users can now drag and drop folders in the left navigation pane into other folders. Users must have "Owner" permissions on both folders.

Note: You cannot drag and drop folders to the root folder. To move folders back to the root folder level, right-click a child folder, select "Move Folder" and select Root from the option list.

Updated Search

Special characters (percent signs, brackets, and underscores) are no longer treated as wildcards.

Password Strength Indicator

Redesigned the password strength indicator.

Column Resizing

Users can now re-size columns on secret grids in the new UI. Column re-sizing is sticky per page across user sessions. Column resizing is not available in Internet Explorer.

Duo Push Notifications

Added option to send secret access request notifications as DUO push notifications instead of emails.

Duo User Preferences

Duo login now remembers user preferences and auto-initiates them (after initial login).

Secret Grid Top-Row Anchors

Added top-row anchors for secret grids.

Home Dashboard Redesign

Redesigned the Home Dashboard page.

Alert Improvements

Redesigned the Inbox and added more alert types (Approvals and Requests, System Alerts, Subscription Alerts).

Domain Name Searches

Added searching for users by domain name.

Secret Audit Access

Audit-only users can now view secret audits on secrets that require checkout or are setup for request access without needing to go through the checkout or access approval workflows.

WEBSERVICES Naming Prefix

Removed the redundant "WEBSERVICES" naming schema from secret audits where the service is also used by the new UI. "WEBSERVICES" still appears in audit reports where the service is not also used by New UI.

Messaging for Permission Changes When Moving Folders

Updated messaging when moving folders so that users are aware of permission changes.

New and Classic UI Feature Sync

We are incrementally porting features from the classic to the new UI. In this upgrade the new UI added support for:

- NATO Phonetics.
- Deleting dependency groups.
- Scrolling for the groups and users picker.
- Secret checkout hooks and corresponding REST API endpoints.
- Showing proxy SSH login credentials information on SSH secrets.
- Setting custom password requirements per secret to override secret template requirements as needed.
- Launcher preferences on the secret-level to override launcher template requirements as needed.
- SSH command restrictions option.
- Launcher settings for SSH launchers to the new UI. Settings not yet implemented include mapped Web launchers, non-mapped Web launchers, and form filler.
- Added a "Save to File" option for audit history and secret grids.

ESXi Support Improvements

Added a configuration option for Secret Server to allow ESXi TLS connections to ignore self-signed certificates, allow certificates from specific issuers (even if issuer is not in trusted certificate lists), or completely skip certificate validation when using ESXi password changer, heartbeat, or discovery.

Important: For security reasons, we do **not** encourage customers to use self-signed certificates. Therefore, the new configuration settings listed below are not accessible through the UI. If you need to alter the default ESXi certificate validation settings, **submit a case through Delinea's Support Portal** for assistance.

New advanced configuration settings include:

- ESXi.IgnoreSelfSignedCerts: If true, ignores any self-signed certs (subject = issuer) from ESXi hosts during heartbeat, RPC, and discovery.
- ESXi.CertIssuersToIgnore: Semi-colon delimited list of issuer names (in format shown on certificate---such as "O=Issuer Name"). Ignores partial chain errors due to certificate being issued by any issuer in this list when that issuer is not in the trusted root or intermediate CAs lists on the server.
- ESXi.IgnoreAllCertErrors: If true, certificate validation will not be performed. All certificate errors will be ignored.
- ESXi.CertificateChainPolicyOptions: Identical to TLS Audit option, but specifically for ESXi. Allows setting X509 options to be applied to certificate validation. This is a comma-delimited list of options.

- ESXi.ClientCertificateIds: identical to TLS Audit option, but specifically for ESXi. If ESXi host requires the client to present a valid certificate, this is a semi-colon delimited list of client certificates on the server to try to present.
- ESXi.AuditTlsErrorsDebug: Identical to TLS Audit option, but specifically for ESXi. If set to true and Secret Server (or DE) auditing is set to DEBUG, detailed debug messages about the certificate chain will be written to the log file.
- ESXi.IgnoreDefaultHostCert: Sets all the TLS configuration options necessary to not fail due to a default ESXi host certificate and its issuer not being in the trusted certificates lists. This is a combination of setting the issuer to ignore and not performing a revocation check. Setting this to true should be the first change to make when attempting to resolve heartbeat, RPC, or discovery issues to ESXi hosts when using PowerCLI versions later than 5.5.

Note: Issues with self-signed certificates previously implemented by customers were caused by a security update to the VMware vSphere PowerCLI in versions after 5.5 that no longer permits the use of self-signed certificates.

Webservices REST API

- The webservices API is now enabled by default for all new Secret Server customers, regardless of licensing. Existing customer settings will not change on upgrade.
- Added a safeguard to the API to prevent license activation attempts when licenses are already activated.
- Added a new REST call to favorite a secret through the API. Favorite a secret by running a POST request to <secret_server_url>/api/v1/secrets/<secretId>/favorite Favorite a secret by setting IsFavorite: Boolean. If left without a value, the favorite status will be toggled. If set to true or false, the favorite status reflects the value of IsFavorite.
- Added a new REST call to get current user's favorite secrets through the API. Get the current user's favorite secrets by running a GET request to <secret_server_url>/api/v1/secrets/favorite An array of secret-related information is returned: {id=1; folderId=4; folderPath=\FavoriteTest; secretName=test1}
- Added refresh tokens for the REST API. Refresh tokens allow users to continue using tools like the Web plugin beyond the normal timeout internal, as long as they remain active. Refresh token expiration is set to "session timeout" value plus 15 minutes. By default, three refreshes are allowed before re-authentication. Enable refresh tokens for Web services under General Configuration
- Added new API Calls:
  - Create Domain: POST /active-directory/domains
  - Create Domain: POST /active-directory/domains
  - Create Script: POST /scripts1
  - Create Secret Dependency Group: POST /secret-dependencies/groups/<secretId>
  - Create Secret hook: POST /secret-detail/<secretId>/hook
  - Delete Secret Hook: DELETE /secret-detail/<secretId>/hook/<hookId>
  - Export Report: POST /reports/export
  - Favorite a Secret: POST /secrets/<secretId>/favorite
  - Get Domain: GET /active-directory/domains/<id>

- Get Domain: GET /active-directory/domains/<id>
- Get Domains: GET /domains
- Get Script: GET /scripts/<id>
- Get Secret hook details: GET /secret-detail/<secretId>/hook/get/<hookId>
- Get Secret Hooks: GET /secret-detail/<secretId>/hooks
- Get secret launcher details: GET /launchers/secret
- Get SSH Proxy Information: POST /secrets/sshproxy
- Launch a secret: POST /launchers/secret
- List a User's Favorite Secrets: GET /secrets/favorite
- Search Domains: GET /active-directory/domains
- Search Domains: GET /active-directory/domains
- Search scripts: GET /scripts1
- Stub Hook: GET /secret-detail/<secretId>/hook/stub/<scriptId>
- Update Secret Hook: PUT /secret-detail/<secretId>/hook/<hookId>

- Updated API calls:
  - Update Secret Field : PUT /secrets/<id>/fields/<slug>
  - Update Secret : PUT /secrets/<id>

Verbose Logging

Enhanced verbose logging for diagnostics to increase clarity and support troubleshooting. This ongoing project began in 10.6.26. Added additional logging for:

- Protocol Handler: Delinea.ProtocolHandler.
- SAML Response Processor: SAML.SamlResponseProcessor.
- SAML Legacy Configuration: SAML.LegacySamlConfiguration.
- Secret Notification Emailer: SecretNotiificationEmailer.
- Event Subscriptions: EventSubscriptions.

General

- Enabled the setting "Enable Local User Password History" to be enabled by default for customers. This blocks a user from re-using an old password when setting a new password.
- Added a configuration option to automatically disable inactive users if they have been inactive for a minimum of one month. Enable this setting if your environment is setup for AD Synchronization. Go to **Admin > Active Directory** in the **Active Directory User Synchronization** section. Click **Advanced (not required) > Set Automatic User Management** and select **yes**.

Bug Fixes

REST API

- Fixed a bug in REST API documentation for recorded session searching to properly detail URL array parameters and display the available values for searchTypes. Also updated the check for searchTypes to be case-insensitive for searching in the documentation. The check for searchTypes is now case-insensitive. Some work was done previously to return an error if no search types were provided. REST documentation was not properly detailing URL array parameters, they are now being displayed correctly and the available values for searchTypes are detailed in the documentation.
- Fixed an issue where creating a folder through the REST API always set the permission inheritance to false on the folder, even when the parent folder was set for inheriting permissions. This meant that upon create, the folder itself inherited permissions from its parent folder but future objects created in the folder did not inherit permissions. The create-folder REST endpoint no longer always copies permissions from the parent folder down on to the sub-folder. Instead, if you set the inherit permissions property in the request to true, permissions will copy down from the parent on to the newly created folder. Should the user omit the inherit permissions property from the request body, the default value is also set to true. In order to **not** inherit permissions from the parent folder for a newly created folder thru the REST API, a user must set the inherit permissions property to false in the request body.
- Fixed a REST API issue where the "View Secret Audit" permission was incorrectly required to access a secret summary through the API. During the permissions check for the REST API, a default route was being taken which included checking for View Secret Audit permissions. A more specific series of permissions checks are now being used which allows the View Secret Audit permission to be ignored.
- Fixed an issue where the search results returned for InheritSecretPermissions in the REST API were not accurate.
- Fixed a bug where REST API calls on systems using Windows Auth would fail after the first call due to logic that incorrectly flagged for cookie expiration.
- Fixed a re-introduced issue where the "Webservice Password Displayed" audit was logged incorrectly when API calls to Get secret fields that were not password-related occurred. This issue was originally fixed in version 10.6.26 but reintroduced in version 10.6.27. The SecretGetQuery.SecretItemsViewed was not being forwarded to the actual GetSecret method which handled the auditing logic. By omitting it, it assumed all secret fields were being viewed by the user hence the password viewed audit was being recorded.
- Fixed a bug where Oauth tokens were deleted for a current user when attempting to change the password of another user through the REST API.
- Fixed a REST API bug where if enableInheritPermissions was set on FolderModel and UpdateFolder objects, updating secrets through the API in those folders did not respect the inherited permissions.
- Fixed an issue where existing file attachments were removed when editing or saving secrets through the REST API.

Mac Launcher

Note: To update your Mac launcher and apply this fix, you must first fully uninstall the Mac protocol handler.

- Fixed a Mac launcher issue where a launched session would hang if user attempted to scroll using a mouse wheel or a track pad. This issue occurred because the 9-bit signed value for handling mouse wheel events was not properly parsed for "coarse-grained" mouse movements within FreeRDP. To update your Mac Launcher and apply this fix, you must first uninstall the Mac Protocol Handler fully.
- Fixed an issue in the Mac Launcher Protocol Handler where FreeRDP was calling to re-sync specific modifiers (Caps Lock, Num Lock) and not the current pressed state of the other modifies on every key-down event during Mac launcher sessions. This caused the other modifiers (such as Shift and Ctrl) to reset back to their base state even when the keys were being pressed.

General

- Fixed an issue where Secret Server could not correctly identify the state of a ServiceNow ticket. Ticket status in ServiceNow is indicated by both a state label (such as New, Open, WorkInProgress, ClosedComplete, ClosedIncomplete, ClosedSkipped, Assess, Authorize, Scheduled, Implement, and Review) and a state value. An update from ServiceNow changed both the labels and the values corresponding to each state, resulting in the issue.

Note: Validating custom ticket states will be addressed in a future release.

- Fixed an issue where heartbeat failed as a bulk operation on ESXi Servers. When a bulk operation ran, multiple threads would try to access the same file on the machine at the same time. A check whether the file was in use was added to resolve this issue.
- Fixed an issue where secret templates containing custom fields caused an error when key management was enabled.
- Fixed an issue on environments with multiple domains where the dropdown for users on the AD synchronization page only displayed members from one domain instead of displaying all users across all domains.
- Fixed a bug where two-factor PIN emails were not sent on initial login if the user's login, the local admin account, and another user were all configured to use the same email address. The SMTP client and application name are now resolved on the initial login page. That way the PIN Code page can read those values and send the e-mail even if the lifetime scope is lost before the PIN Code page is loaded.
- Updated some instances where a browser's built-in auto-complete feature would incorrectly enter data into fields when a Secret Server page was loaded. The root of this issue was that the mechanism for disabling a browser's auto-population for username and password fields on page load changed across all browsers. The old mechanism added the attribute autocomplete="off" to the target input or form field if you did not want the browser to auto-populate on page load with user credentials. The new way changed the value of the added attribute to be autocomplete="new-password" for populated fields.

Note: Clear browser history cache to apply these fixes.

- Fixed a bug where changes were saved even after canceling on the **Admin > Configuration > Login** tab.
- Fixed an issue where checkout and check-in hooks caused an exception error for a user if they only had list permissions on a privileged account when attempting checkout. This issue occurred when attempting to load the privileged secret with the user's permissions, which would fail unless the user had view or greater permissions on the privileged secret. A change was made to load the privileged secret using the system identity to prevent access issues.

- Updated extensible discovery for the PowerShell scanner to automatically fill in an AdGuid and DistinguishedName when those values are not present on the scan item for OU/Host Ranges and Machines. If scanning Active Directory, adding DistinguishedName to the scan item is recommended. In this instance, you must remove the domain portion of the DistinguishedName.
- Fixed an issue where saving a secret policy on a new secret template did not allow the latest password changer templates to be selected for password rotation. This prevented secrets of the latest template types to be assigned as a privileged account for the password rotation.
- Updated the group picker for editing users in groups so that users are now sorted consistently via alphabetical ordering. Before this update, usernames in the two sides of the group picker were sorted differently. Updated all of these, Group Create, Edit and View pages, to sort by the "Known As" field. If DomainId is null, the field returns DisplayName. Otherwise, it returns DomainName\DisplayName.
- Fixed an issue where Delinea One (T1) users were unable to login to Secret Server through the mobile and desktop app when Delinea One was enabled for Secret Server. There's a new configuration option "Enable Delinea One Integration" in Secret Server to do API authentication against Delinea One (the old way) or not (the new way). This is under **Configuration > Login**. When checked, the mobile app should accept Delinea One credentials. When unchecked, the mobile app should accept local account credentials. If T1 is turned off, or if the user doesn't have a T1 mapping, it will always accept local account credentials.
- Fixed a bug that blocked workflows from being enabled on a secret in the new UI for Cloud Professional users if no approvers were defined on the workflow. When no approvers were defined for a workflow step an error was thrown trying to retrieve the max number that could be set for required approvers at that step.
- Fixed an issue where custom reports that contain "Permissions" as a field name were not successfully created.
- Fixed an issue where disabling two-factor authentication using the "Lost Phone" option did not trigger a "Two-Factor Changed" event notification. The event logs did not track resetting 2FA auth. Added two new events, one for a successful TOTP reset and one for a failed TOTP reset.
- Fixed a bug where the "Last Scanned Date" for AWS and domain accounts were not properly updated on the Discovery Network View. Updated the Last Polled Date for the Domain Accounts to use the LastDiscoveryCompletedDate from the tbDiscoveryConfiguration table.
- Updated error messaging when Secret Server is unable to connect for heartbeat status. Added exception handling for Socket timeouts and errors to be treated as unable to connect instead of as an Unknown error.
- Fixed a bug where custom launchers that passed the port field from a secret as a launcher argument would fail because the launcher was using the port value from the secret template launcher configuration instead of the port value from the secret. For non-proxied custom launchers, this caused the port argument to always be zero.
- Fixed an issue where users synchronized from Active Directory through Distributed Engine did not have their UserPrincipalName populated.
- Fixed a bug where disabling users for an event subscription blocked the event subscription from being saved. This issue was related to a Linq expression which used Single instead of SingleOrDefault, which caused the respective exception when no elements were returned from collection.
- Fixed an issue where creating sub-folders from the folder menu did not use "No Policy" default settings when set to not inherit a policy.

- Resolved an application error that occurred when the expiration interval was set too high on a Secret Template. When the expiration was set to a large number the date calculation would overflow causing both secret search and secret view to break since they both tried to calculate it. Now both cap the expiration days at 9999 as the UI currently limits expiration to that maximum.
- Fixed an issue in the new UI where deleted secrets remained on display on the Favorites page after a page refresh.
- Fixed an issue in the new UI that occurred when fields in a column in a secret grid view could appear blank if a secret template had multiple fields with the same name.
- Fixed an issue in the new UI where folder breadcrumbs were incorrectly sorted by Folder ID instead of by the location in the folder tree. Folder breadcrumbs were being incorrectly sorted by ID on the server before the API response was generated, which only works if no folders have ever been moved. That ordering was removed and now the folder order is returned properly from the API.
- Updated tooltip messaging for the user interface configuration options.
- Fixed an issue in the new UI where users were not always directed to the All Secrets page upon login. When a user was logged out in angular, the ReturnUrl was not always set. The ReturnUrl query string is used after logging in to know where to redirect a user.
- Updated the new UI to hide reports under admin options when a basic user does not have the "View User Audit Report" permission.
- Fixed a Firefox browser issue for versions 60.0 and greater where search prompted errors in the new UI. A nonstandard JS property was being used that isn't supported in all browsers. Property was replaced with its standard equivalent to ensure support in all browsers.
- Fixed an issue where the password field did not re-appear after removing the "hide launcher password" policy on a secret.
- Fixed an issue in the new UI where Danish Characters did not display correctly in the Folder list.
- Fixed an issue where the Heartbeat Status column did not always display on page load in the new UI.
- Fixed a bug where allowlisted launcher options did not properly populate in the new UI.
- Fixed a bug where the password history for a secret did not display the correct password if the password included a less than symbol (<).
- Fixed an issue in the new UI where Approval, Access Request, and Checkout secrets incorrectly blocked access from administrators when Unlimited Admin Mode was enabled. State service was returning state values without considering unlimited admin mode.
- Fixed an issue where read-only mode was set as default in the new UI until the interface fully loaded, resulting in brief displays of the Read-Only mode notification if a user's browser experienced slow load times.
- Fixed an issue where editing a secret policy would throw a null ref error in the new UI if the policy applied to a folder but was not enforced.
- Fixed an issue where discovery was unable to recognize special characters (such as Ã, ‡, ƒ, and ±) in organizational unit (OU) objects when scanning an entire domain.
- Fixed an issue with Active Directory synchronization where using the SynchronizeNow group flag could cause users not in that group to be disabled when AD sync was set to "Mirror AD."

- Optimized SQL query used to process large result sets in the tbSecretPasswordResetSecrets table. This addresses an issue where expired secrets failed to process due to large volumes of secrets.
- Fixed a bug where domains listed with friendly names did not adequately check for uniqueness of Fully Qualified Domain Names (FQDNs) therefore allowing FQDN duplication. If FQDN duplication occurred, AD Synchronization domain mismatch failures resulted when running AD Sync.
- Enhanced performance for discovery when one endpoint returns large numbers of accounts (20k+).
- Fixed an issue with search in the new UI where searches were locally cached instead of cleared on route navigation within the app.
- Updated the new UI to accept non-ASCII usernames.
- Fixed an issue where the Reports page in the new UI did not process HTML-encoded data, leading to special characters like "é" not displaying correctly on the page.
- Fixed an issue in the new UI where the "Save to File" option on tables was not created in a download method that Internet Explorer 11 supported.
- Fixed an issue where copy and convert actions on a secret template could apply parent folder policies to the new secret rather than applying the original secret's policy. This occurred due to using a cached version of the secret template instead of the updated version.
- Fixed an issue where non-standard UTF8 characters were not saving properly when saving reports to file in the new UI.
- Fixed an ordering issue when querying reports where the user-identification audit ran after the report query instead of before.
- Fixed an issue where live view-session videos did not work in the Internet Explorer 11.
- Fixed an issue where outbound queue messages overloaded in some environments. Added expiration time to outbound discovery and Active Directory sync messages. AD sync and discovery scan messages now expire based on their configured interval.
- Fixed an issue where the bulk conversions for secret templates did not populate the target template list.
- Fixed an issue where naming patterns were not properly enforced in the new UI when creating secrets.
- Fixed an issue where the approval-request expiration time did not properly handle time zone differences. We changed validation to compare minutes instead of days.
- Fixed a bug where the "confirm action" button did not activate on bulk actions when assigning the "inheriting permissions" action.
- Fixed an issue where changed fields on the edit page for assigning secret auto-schedules could block a schedule from saving.
- Resolved issue with SSH key password rotation for some versions of Unix and specific templates.
- Fixed issue with IAM Token rotation over distributed engines.

Secret Server Cloud: Legacy Release Notes

Note: This document covers releases prior to 9/21/2019.

Cloud Release 6/15/2019

Security Advisory

Cloud instances are no longer vulnerable to this issue.

Upgrade Notes

New UI Wizard

After June's Cloud upgrade, the first admin who logs in to Secret Server Cloud (SSC) is prompted with a wizard that walks through the new user interface (UI) configuration settings for their instance. Users can adjust these settings at any time from Admin > Configuration in the User Interface section.

Radius IP Addresses

Important Note: IP addresses used by SSC for Radius clients will change during this release.

If you use Radius with SSC and have firewall rules or Radius configuration that specifies the SSC IP addresses, then you must update those rules prior to June 15th to maintain continuity of service. The new IP addresses will be active after the upgrade on June 15th:

- US Cloud: 40.76.197.147 and 40.121.181.52
- EU Cloud: 51.4.141.94 and 51.4.194.120
- AU Cloud: 20.36.47.199 and 20.36.45.106

Zero Downtime for Cloud Upgrades

Scheduled releases no longer block access to the website during the upgrade window. The SSC UI remains accessible during the migration process as customers are moved to the new version. A customer's typical migration period is 5 to 15 minutes.

New Features

New User Interface

We are incrementally introducing feature parity between the new and classic UIs. In this upgrade the new UI added support for:

- Emailing reports to users.
- Deleting reports.
- Converting secret templates.
- Uploading files to secrets.
- Copying and converting file attachments.
- Setting automatic password change rotations when creating new secrets.
- Rotating SSH Keys and Passphrases for SSH secrets.
- Rotating SSH Keys and SSH passwords using a bulk operation.

- Displaying custom columns in the folder view.
- Updated the Secret Picker to include a folder tree browser when selecting a privileged secret for Remote Password Changing.
- Added a password complexity indicator that displays when editing a Secret's password.
- Updated Search functionality to include folders.
- Users can now "favorite" folders by clicking stars on folders in a table view or by right-clicking folders in the left navigation pane.
- Additional "View" audits were added to the new UI to track users who view Next Passwords, SSH Keys, and Passphrases.
- Optimized SQL queries when searching for secrets in the new UI. This includes:
  - Faster searches, the paging and sorting is now performed in the database instead of the web server
  - Ability to filter by secret permission, extended type id, password type id, RPC enabled, recently used date range
  - Can now return extended fields
  - Sorting by an invalid field now returns an exception instead of sorting by the secret ID

Advanced Session Recording Without Launcher

- Added video recording capability to the Advanced Session Recording agent, with a new configuration setting that can enable recording of a video on an endpoint even when a launcher is not used to begin a session.
- Advanced "headless" session recordings will show a 'Session Minimized' notification for periods during which the session was minimized, or otherwise hidden from the user such as during the final stages of logging out.
- To prevent unnecessarily long session recording sessions, added a "Max Session Length" setting for recorded videos. The default for Cloud users is 8 hours. Administrators can set this to a maximum of 12 hours.

Secret Templates

- Added a new "No Password" template for SSH/Linux Key rotation. These templates allow users to rotate SSH and Linux keys without the password field requirement that exists on other Unix secret templates in Secret Server. When creating a new secret, select Unix Account (SSH Key Rotation - No Password) as the template to use this new feature.
- Added a native Amazon (IAM key) rotation template. When creating a new secret, select Amazon IAM Key as the Template to use this new feature.
- Updated launcher and password changing for the IBM iSeries secret template.

Integrations

Added a new biometrics eWBM integration for FIDO2 authentication with Secret Server. eWBM's website at http://www.e-wbm.com/. A connector application is now available to link Secret Server to identity management tools using the System for Cross-Domain Identity Management (SCIM) standard. For further information please see the SCIM Connector documentation:

- Install Guide
- Getting Started Guide

Enhancements

Session Recording

- Updated the "View Session Data" feature for session recording to keep users from viewing session data on any ongoing session.
- Added searching by date and time for session recording videos.
- Updated the advanced session recording agent (ASRA) so that the SecretLauncherSessionMatchingService can now match the ShortName to the NETBIOS name of domain objects in Secret Server. Windows terminal sessions report the pre-2000 domain NetBIOS name, which is relayed from the ASR agent to Secret Server and used for matching Windows terminal sessions to secret launcher sessions. This is an issue if the domain on the secret is the domain FQDN or friendly name and matching fails. For example:
  - You have the ASR agent installed on a machine which is joined to a domain (mydomain.com) with a pre-2000 NetBIOS domain name "SOMETHINGELSE"
  - Your secret has the domain name "mydomain.com"

  When you launch into the machine, Secret Server attempts to match "mydomain.com" (from the secret) with "SOMETHINGELSE" (the domain NetBIOS which the ASR agent relays) and fails. Secret Server resolves this issue by trying multiple methods to get the appropriate FQDN for the Windows terminal session and using the best result from those methods, which should result in matching correctly more frequently.
- Updated advanced session recording matching by automatically mapping to the secret template's RDP fields automatically. Matching was improved by checking the secret template's RDP launcher mappings. Now, no matter what the fields are named on the secret template, the correct username and domain fields are used, based on the launcher mappings.
- Advanced session recording can now match sessions using DNS resolution. When using an RDP launcher on a secret with a machine or computer name, instead of an IP address, the protocol handler does a DNS resolution of the hostname and reports the IP back to Secret Server, which is saved for the entry in tbSecretSession. This helps advanced session recording find additional matches when the hostname does not match the computer name, only if the hostname resolves to an IP address found directly on the target computer. If there is any NAT used, this will still fail to find a match because the NAT IP is not found on the target computer.

Secret Folder Import and Export

- Added subfolder creation during secret importation.

Performance

- Enhanced performance of request/response time for SecretsController API methods by resolving Autofac dependency chain issues.

Search

Added default index separators to include the following characters:

- ? Question mark
- ! Exclamation point
- @ At symbol
- # Pound symbol
- (left parenthesis
-) right parenthesis
- [left square bracket
-] right square bracket
- { left brace
- } right brace
- ' apostrophe
- " double quotation mark
- - hyphen

Configure indexing separators at Admin > Search Indexer.

Discovery

- Updated the ESX discovery scanner to updated versions of the VMWare/VSphere SDK. Logs
- Enhanced verbose logging. Added additional logging for the Scheduled Task Scanner, Discovery Classes, Active Directory, and Password Changer logs.

Bug Fixes

Launchers

- Fixed an issue where the Mac launcher was not automatically populating unique password prompts when connecting with PuTTY.
- Fixed an issue where the Custom Launcher screen resolution setting for Mac Launchers was not properly applied. Previously, custom launcher sessions used full-screen resolution. Note: If there are no custom values set, session resolution will default to 1024×768.
- Fixed an issue where using a Windows account launcher with a domain account would fail if the launcher was configured with an IP address in the machine field. The logic detected the computer name as an IP address and then returned before setting the domain field.
- Updated the Mac Launcher to hide the session recording indicator for users when Configuration > Session Recording > Hide Recording Indicator is enabled for the launcher. The mac launcher now uses the "hideRecordingIndicator" variable stored in the sessionInfo object. Initially, there was no logic involving hiding the indicator. Added the "hideRecordingInidicators" function to the object ALaunchedTask. Updated logic under "refreshRecordingIndicator" under TaskMonitor to call "hideRecordingIndicators" when determining if it should show any icon for active watchdogs. Updated watchdog to contain a public property that is passed on

instantiation--"hideWatchdogMessages". On watchdog start or stop, it now validates the "hideWatchdogMessages" is false before displaying alerts about the session being recorded.

- Fixed a custom URL issue where launcher images did not properly display when accessing Secret Server from different sites. The custom URL was a different sub-domain from the URL that the site was accessed from. The images were served from the custom URL and our content security policy was blocking those requests. The content security policy is used to prevent cross-site scripting attacks. The solution uses the URL given instead of the custom URL.
- Updated the Mac launcher to include support of TLS versions 1.2 and 1.3.
- Changed the default launcher setting for RDPUseComputerForDomain to "True" to allow the launcher to work on all Windows versions currently supported by Microsoft upon install. This was tested on Windows 8.1, 10, Server 2008, Server 2012, Server 2016, as well as on MacOS 10 Sierra, High Sierra, and Mojave. It prevents launchers from failing unintentionally when connected to some Windows OS versions.
- Updated launchers to accept upper case characters in URL strings on a Web password secret. Previously, launchers set all uppercase characters to lowercase before launching, which caused launch failures for URLs that are case sensitive.
- Fixed an issue where a user launched a session recording but then immediately exited, Secret Server still attempted to record the session, resulting in inaccessible videos that were 00:00 seconds long. To resolve, zero-length sessions are no longer turned into videos and now returns the error: "Session too short - no video."

New User Interface

- Fixed a bug in Internet Explorer for the new UI where the Download File button threw an "access denied" error.
- Fixed a bug where the dependency log view did not sort correctly on the Date Recorded column.
- Fixed a bug in the new UI where users with "edit" and "administer folders" permissions were unable to create subfolders within a parent folder.
- Fixed a bug in the new UI where users without the "Assign Secret Policy" role permission were unable to create new secrets in a folder. The internal logic for validating whether a user may assign a secret policy to a secret did not consider the special case of secret creation. The user's personal permission to assign a secret policy should be ignored during the creation process if they did not assign one, but the code acted as if the user had assigned the policy, not that the policy was assigned by the folder.
- Fixed a bug in the new UI where breadcrumbs for the Workflow and Teams audit pages displayed "audit" instead of the component's name. This issue also caused breadcrumbs to reload when switching between tabs on the page.

Time Conversion

- Fixed a date-time issue where abbreviations for the month of the year in languages other than English were not properly logging to Syslog.
- Fixed an issue where records in custom reports were logging as the UTC time zone

Scripting

- Updated REST API call "Get Folder Permissions" parameter for filter.userId to return all user permission information.
- Added enhanced error logging for Oracle scripts.
- Fixed a bug where the API incorrectly required Owner permissions on a secret to change the password on that secret. Edit permissions are now required when changing a password through the API.
- Fixed an issue where the "Webservice Password Displayed" audit was logged incorrectly when non-password-related API calls to "Get Secret Fields" occurred. The REST endpoint GET /secrets/{id}/fields/{slug}, when used to view a non-password secret's field was incorrectly recording an audit stating "WEBSERVICE PASSWORD DISPLAYED." This was due to the endpoint's implementation using the existing GetSecret REST logic, which manages the creation of this audit. This logic assumed you would respond or display to the user with the secret's fields (items and data) when getting a secret that contains a password, recording an audit. To correct this, the GetSecret logic now supports an optional parameter that specifies which of the secret's fields (items or data) you will be responding with (displaying) to the user. If the items supplied to this method do not contain a password field, then the password displayed audit is not recorded. The corresponding GET /secrets/{id}/fields/{slug} endpoint was then updated to supply this method with the requested secret field, fixing this bug.
- Fixed a bug where Web services incorrectly returned SecretDependencyTemplateIDs as null. The SecretDependencyTemplateId is now returned with every dependency found by GetDependencies in SSWebServiceHandler.
- Fixed a bug where an "Application Account License limit has been reached" error was incorrectly displayed after editing existing application accounts, instead of only when creating new user accounts. This bug was due to a flaw in the user license logic where application account licenses were incorrectly returning the API_MetApplicationAPILimit error code when editing and attempting to save an existing application account user, while at the current application account user limit. Specifically, the logic assumed the addition of a new application account user--it did not support editing an existing user. The error occurred at the application account user limit where adding a new user is not permitted. The fix was to check if users were adding a new application account user or editing an existing user when determining whether to return the error code, API_MetApplicationAPILimit.
- Fixed a bug where API group membership changes were not properly audited. This bug was due to a lack of auditing logic in the CreateGroupUser and DeleteGroupUser REST endpoints. The fix was to observe what audits the UI were doing upon adding and removing users to and from groups, then applying that same logic to the CreateGroupUser and DeleteGroupUser REST endpoints. Some extra logic was added to prevent recording audits if no operation was executed, such as moving a user to a group it already belonged to or removing a member from a group it was not a member of.
- Fixed a bug where users were unable to create subfolders that inherit folder permissions through the REST API, despite the user having "Edit Folder" permission.
- Updated REST API documentation for SSC.

Authentication

- Fixed a SAML login issue that could cause login failures if multiple users had the same User Principle Name (UPN) in Secret Server. The account that was created first would be prompted for SAML authentication even if

that account was in a disabled state. This caused login failures with an error saying the user's account was disabled. Updated GetUserByUserPrincipalName to return a user only if there was exactly one user found with that UPN that was active. If there is not exactly one, it returns null. Handled a few logic flows where a GetUserByUserPrincipalName request is followed with a request to GetUserbyUsername. In cases where a valid UPN is entered, an error is produced at GetUserByUsername. These errors are now handled and logged. Secret Server.

- Fixed a bug where if an instance had two users with the same username (one active user, one disabled user) across different domains two-factor authentication (2FA) enabled for login, a user trying to reset their login might get a validation error because the reset code they entered would be checked against the code of the other user with the same name. Secret Server now checks the domain of the user who is trying to login when determining which user account should be used for the 2FA reset request. This removes ambiguity when users in different domains have the same username, thus ensuring the correct 2FA validation is used.
- Fixed an issue where 2FA failures were not showing up in the "Failed login attempts" report. The 2FA failure message was different from a normal login failure message. The report SQL was changed to display any login failure message.
- Corrected a configuration error that changed the 'Disable Radius NAS-IP-Address Attribute' setting to have the opposite effect. The logic using the DisableRadiusNasIpAddressAttribute setting was set to return if false instead of true. This was an old application setting that would normally be false for most customers. When it was added to the UI configuration pages, the setting was implemented incorrectly, reversing the logic. In versions 10.5.14, 10.6 and 10.6.1, it is being disabled by default due to the backwards logic. This release corrects the configuration setting to work as described in the UI.
- Fixed a bug where the authentication function for the "Attempt User Password" Radius setting did not work when using the REST API.

Discovery

- Updated the PowerShell local account and dependency scanners to timeout after a set timeout. The application settings DiscoveryScannerTimeout and DependencySearchTimeout have been moved to settings on the individual scanners. If this setting was only set in an application setting, it needed to be reset on the scanner setting. Wrapped the PowerShell dependency and local account scanners in a System.Task to cancel after a timeout period.
- Fixed an issue where out-of-the box dependency scanners (scheduled tasks, application pools, and service accounts) were not correctly copying as the same specified scanner's type.
- Fixed an issue where discovery imported group-managed service accounts (gMSAs) as new computers and subsequently attempted to scan them for accounts.

Password Changing

- Fixed an issue with Oracle privileged password changers. Before this update, Oracle password changing logic for privileged accounts was not correctly using the designated privileged secret to perform password rotations, leading to failed password change attempts.
- Fixed an issue where missing parameters in the Sonic Wall password changer resulted in multi-factor-authentication (MFA) failure. The info sent to the Sonic Wall password changer now includes default values for MFA code and future MFA codes so those do not reference null. The PhantomJS script for Sonic Wall now looks

for the host name at a different place in the parameters list.

- Fixed a bug where the IBMi mainframe password changer timed out during process cleanup. IBM iSeries password changer has two threads during a session, the primary thread processing the session or request and a subsequent one that is spooled up during the process to ensure the interface framework (ws3270) does not get stuck in the wait state when the wait command is called. The second thread may detect false stuck wait states and attempt to close the session. If this happens after the primary session has completed the wait and has already closed, an "Access Denied" exception occurs. That exception can still be due to permissions, so it is worth checking, but most of the time this is the error thrown by the .Net Process framework when attempting to exit a process that has already exited.
- Resolved issues with error handling, command pause handling, and session termination/logoff when using remote password changing (RPC) with IBM z/OS and IBM iSeries. IBM z/OS and IBM iSeries use the same base architecture within RPC to manage and execute commands. In some cases, the password change process did not correctly resume after a pause command. In others, the pause did not take effect, causing commands to execute prematurely and showing incorrect failures. Code was also added to clean up failed RPC/heartbeats once a user is logged in. An identifier to the logoff command now informs RPC of the correct command to kill sessions in cases where it needs to terminate prematurely.
- Fixed an issue after a Google update where changing Google account passwords returned incorrect messaging whether a password change failed or succeeded.

  Important Notes:

  - The Google RPC only works if the user has previously logged into the Google account from the IP address that is trying to change it. In other words, the user needs to log into their Google account from a Web browser and allow the sign-in from an unrecognized IP address before attempting to rotate the password from that IP.
  - If there are too many failed login attempts, Google may block the login for some time.
  - If Google prompts a reCAPTCHA field to verify a human is changing the password, the RPC for the Google account will fail.

  Technical Details:

  Updated the URL check for the password changed test to fix the failed password change when it was a success. Updated the missing error code to fix the successful password change when it was failing. Also added better error checks to inform the user of the specific password change error.
- Added logging to provide details when password change attempts for Google accounts fail, including:
  - NewPasswordLengthTooShort
  - NewPasswordLengthTooLong
  - InvalidCharactersInNewPassword
  - NewPasswordNotComplex
  - IsAPreviousPassword
  - SameAsCurrentPassword

Secret Import and Export

- Fixed an issue where a user was able to edit the error message in the "Import Secrets' dialog box after importing a secret.
- Fixed an issue where secrets were not displaying for users when a left bracket appeared in the parent folder name. We replaced any instance of [in a LIKE clause with [[] The first [opens the character comparison list. The second [is the character we allow for comparison. Finally, the closing] closes the character comparison list. It is unnecessary to replace additional instances of] because they mean nothing without a matching [
- Fixed an exportation issue where the double quotation mark (") was not included in CSV files.

Security

- Resolved a security issue in the Protocol Handler.
- Fixed a security issue where users retained Unlimited Admin permissions for a short time after the Unlimited Admin Mode was turned off, due to a caching issue in the background worker. The issue was related to caching configuration data for 5 minutes and how the cache was being managed by the ServiceLocator used by the background worker. Turning Unlimited Admin Mode on or off would clear the configuration from the cache used by the UI, but the background worker had its own cache, which did not get cleared. The fix was to reinitialize the cache provider on the ServiceLocator whenever starting a bulk operation or secret importation to ensure these functions get a fresh copy of the configuration and Unlimited Admin setting.
- Resolved inconsistent behaviors in role-based permissions.

Other Bug Fixes

- Updated session recording to address an issue where video recordings intermittently returned black frames during the session's video playback. This was observed on older laptops running Windows 7 (either 32-bit or 64-bit) with integrated Intel graphics. Neither metadata nor keystroke logging (for advanced session recordings) were affected by this issue, only the viewing experience of the video, which could be interrupted by black frames. The fix was to use a different Windows API call as a fallback whenever a black frame is returned from the original screen capture attempt. A drawback to this approach is that fallback API simply captures the whole window including whatever might be on top of it. To counter this, logic was added to ensure that the RDP window is topmost during image capture. There remains a small possibility of capturing an isolated black frame or two if another window is moved on top of the maximized RDP window when an image is captured.
- Fixed an issue where some Secret Server background threads attempted to perform tasks when Secret Server was disconnected from its database. The issue was that some background threads would exit their continuous monitoring loop. Specific conditions caused new SQL server connections to time out, but the open connections continued working. This resulted in Secret Server working normally but caused the monitoring loop to exit and fail to restart, breaking the monitoring of the thread. This was fixed by catching the error that caused the loop's exit, allowing the monitoring process to continue.
- Fixed a bug where removing an event subscription item deleted the wrong item. When editing the events to trigger the event subscription, the ID of the event stored in the database identified which event to remove when an item was deleted. If an item was added, but the updated event subscription had not been saved, that ID was not generated yet. This caused the first event in the list to be deleted every time a newly added event was removed while editing. A temporary ID is now generated when a new item is added to the list so that if the item has not been saved to the database, the correct item in the list can now be identified.

- Resolved an issue where backups were unable to complete if a Web application file was in use by a worker process. When looping through the files to be backed up, we now catch errors for any files that cannot be accessed by the backup process due to being locked by another process. When an error is caught, the issue is logged, and the backup process continues to the next file.
- Fixed a bug in the old UI where a Safari browser repeatedly prompted Mac users to download and install the plug-in when attempting a copy-to-clipboard action. The old UI prompts for our Safari plug-in to be downloaded for using the user's clipboard. This is no longer used in the new UI. The link to the plug-in was removed. Now, the user loads the password by clicking the "load" image and then can copy the secret's value to the clipboard by clicking the copy image.
- Fixed an issue where a Distributed Engine would return "Unknown Error" for a secret heartbeat if there was a transient error publishing the real secret heartbeat result back to Secret Server.
- Fixed a bug where sorting groups by the "Created" field did not properly order groups in the table.
- Improved page size restrictions when assigning roles and editing groups for customers with large numbers of users.
- Fixed a bug where searching from a tab did not correctly output search results until the user navigated back to the Home Dashboard. To fix this bug, users now are automatically redirected to the Home tab when performing a search.
- Fixed a bug that allowed Secret Server Professional Edition users to set a Workflow Access Request secret policy that should be unavailable in Professional Edition, causing an error when adding secrets to folders using that policy.
- Lengthened the default maximum value for the transaction timeout on the installer from 10 to 90 minutes to prevent installs from failing due to longer database setup. The issue was caused by lengthy transaction times when running the installer against an Azure SQL database. The default maximum value for a transaction timeout is 10 minutes. We added code to allow us to make the transaction larger and set the timeout to 90 minutes.
- Fixed a bug where converting a secret to a different template created two active copies of the secret: one converted, one unconverted. Inside the SecretDataCopier, when setting permissions, the code did not know it was due to a conversion. It then copied the folder permissions down onto the secret, instead of using the secret permissions that were on the secret originally. This meant that in the scenario above, the user did not have edit rights, throwing an "Access Denied" error.
- Fixed an issue where launcher-based sessions were not displaying proper messaging during session recording. Message boxes are now correctly bound to their parent process (the launched process), and message box behavior has changed. Before this update, message boxes launched without a proper parent process would be minimized and would not display or block input if you restored the launched process, because they did not belong to the launched process. After this update, message boxes are launched with a proper parent process always appearing on top of that process and preventing input to that process until the message box is acknowledged.
- Fixed a bug where Distributed Engines configured to callback to multiple Web servers did not work if the server names were separated by semicolons.
- Fixed an issue where after an SSH key expiration occurred, uploading a new key remained in an expired state instead of properly updating the key.

- Resolved an exception error that occurred when closing an RDPWin session window after a secret-checkout session ended.
- Fixed an issue in customer environments configured with ticketing systems where approving a request from an event notification email caused an exception error to occur, rather than completing the approval.
- Fixed an issue where reports containing the #CUSTOMTEXT dynamic parameter failed to send emails.
- Fixed an issue where secret templates containing custom fields caused an error when Key Management was enabled.
- Fixed an issue where creating sub-folders from the folder menu did not use "No Policy" default settings when set to not inherit a policy.

Cloud Release Date 3/9/2019

This Secret Server Cloud (SSC) release corresponds loosely with the on-prem version 10.6.00000. The release date varies with location. First release is scheduled for March 9, 2019.

Upgrade Notes

Installing the New Advanced Session Recording Agent

Customers using Advanced Session Recording need to deploy the new agent once their SSC instance upgrades to Secret Server 10.6. Secret Server 10.6 does not support the RDP Monitoring Agent from Privilege Manager for recording keystrokes or process auditing. For details, please see the Secret Server Advanced Session-Recording Agent Installation Guide.

New Features

New User Interface

SSC offers a new user interface (UI) with a redesigned left navigation panel and other improvements. New SSC users default to the new UI on first access to SSC 10.6, and existing users can enable the new UI as desired. For an overview of the SSC UI.

Workflows

- Added a new enterprise feature--multi-tiered secret approvals, which grants secret access after navigating multiple approval layers. This feature is available in Secret Server's new UI only.
- Added two new Roles to support multi-tiered secret approval workflows:
- Workflow Administrator–can administer workflow permissions.
- Workflow Designer–can create new workflows.

Advanced Session Recording

- Enhanced speed and performance for both basic and advanced session recording.
- Added a new agent to Secret Server for advanced session recording that captures metadata from launcher sessions to targets. Secret Server

- Customers using advanced session recording must deploy the new agent when upgrading to Secret Server 10.6. Secret Server 10.6 does not support the RDP monitoring agent from Privilege Manager for recording keystroke or processing start data.
- The new advanced session recording agents uninstall themselves when deactivated from SS.
- Secret Server must have appropriate permissions required to access all file paths listed for saving the configuration and recordings.
- When creating a new site, a specific site-to-folder-path relationship will not be created automatically. Instead, the secret will use the default path (whatever path is used for the local site). When you edit, a row will show up with the default value per site already loaded.
- Added a warning for session recordings experiencing many unprocessed videos.

Teams

- Added a new "teams" feature so that Secret Server administrators can segment users and groups within one Secret Server instance.
- Designed as a convenience feature, users that lack an elevated role permission within a team cannot select users, groups, or sites that exists outside their team, including in dropdown options and searches.
- Added three new roles to support teams:
- Administer Teams--Can create, edit, and view teams.
- Unrestricted by Teams--Can view all users, groups, and sites. The default User role in Secret Server now has this permission, so role permission customization is needed for teams to take effect.
- View Teams--Can view all teams.

FIDO2 (YubiKey) Authentication

Added a new integration with Yubico and other FIDO2 implementations. Secret Server can now be configured to use FIDO2 tokens (YubiKeys) as a method for multi-factor authentication.

Launcher Compatibility

Added backwards and forwards compatibility to the Secret Server launcher protocol handler.

Telemetry

Added a new feature that sends anonymized usage data to guide future research and development plans at Delinea.

Remote Password Changing

- The Secret Server API can now conduct remote password changing on AWS secrets. That is, the API now supports PowerShell script password and dependency changes for AWS IAM token rotation. Generated values are passed back for saving on the secret. This is useful for tokens generated by an external system during rotation. This is only the underlying architecture for use via PowerShell scripts, not a full password changer.

- Added a new IBM iSeries password changer template to enhance 5280/IBM Series 7.1-7.3 systems support including features such as program functions.
- Added a $$Pause command for the Custom SSH password changer so that administrators can prevent run commands executing immediately after login, which can cause failed executions.
- Added support for VMware password changing and discovery to work with PowerCLI up to version 10.1.1.
- Increased default RPC retry interval to run every 15 minutes and to cap at 10,000 consecutive tries. "Unlimited max attempts" is no longer an available option. Retry Interval can be manually configured by 5-minute increments. Heartbeat interval is also now capped at minimum of 15 minutes.

SDK

A new SDK version is on nuget.org that supports targeting for .NET Framework 4.5.1 in all packages. The only external dependency is Newtonsoft.Json (v11.0.2). See https://www.nuget.org/packages/Thycotic.SecretServer.Sdk.

RabbitMQ Helper

Added federation and clustering support for RabbitMQ Helper:

- Federation: https://thycotic.github.io/rabbitmq-helper/usecases/federation/
- Clustering: https://thycotic.github.io/rabbitmq-helper/usecases/clustering/

Enhancements

- Added the ability to download a recorded session with the API. This new API call will not download metadata and only applies to basic session recordings. The call is api/v1/recorded-sessions/{id}/session-recordings.
- Added a PrefetchCount application setting to allow customization of engine response message processing and to enhance processing speed.
- Enhanced load performance for discovery.
- Added a cache for session configuration to prevent excessive callbacks to SessionRecordingConfigurationProvider.Load during discovery scanning.
- Added a new setting for configuring the SSH RPC timeout interval to all applicable SSH secret template settings pages. Prior to this fix, users were not notified when a group name exceeded the maximum character length and instead experienced a web session hang.
- Added a setting to discovery that can check whether IIS is installed before scanning for application pools. To enable the "Verify IIS is Installed" setting:
  1. Navigate to Admin > Discovery.
  2. Click the Edit Discovery Sources button.
  3. Click the link for the desired source. Its page appears.
  4. Scroll down to the Find Dependencies section.
  5. Click the edit (pencil) icon next to the Application Pool Scanner entry.

In extensible discovery, scanners are setup to run in consecutive order across many organizational units within large environments. This setting was added to allow Secret Server to skip the process of scanning for application pools whenever a machine does not have IIS already installed to enhance performance and cut down on run times. Secret Server

- Added support for a RADIUS challenge in web services. Secret Server will now return a "RadiusUSAccessChallenge" error if an additional prompt is needed. To use this functionality, on-premises Secret Server needs to connect to the same node on both authentication calls. Previously, Secret Server could only handle a single request from the RADIUS authentication process. This enhancement uses caching, so authenticating scripts need to hit the same web node to use a challenge authentication. This fails when using REST + Secret Server On-Prem + Load Balancing + RADIUS Challenge Authentication combined. The workaround is to hardcode REST scripts by IP or FQDN.
- Added various enhancements to the Upgrader. Enhancements will not take effect until your next upgrading process.
- Upgrade enhancements include updated logging, the removal of unchanged files during the upgrade process, re-ordering of tasks to improve performance, and enhanced messaging.
- Enhanced error messaging when heartbeats fail due to an unavailable machine state. Prior to this enhancement, when heartbeats failed on a machine due to disconnection it flagged an "Unknown Error." Now, the machine will return an "Unable to Connect" status.
- Added and enhanced the Custom SSH password changer's console output logging for debugging. Prior to this issue, command set logging was removed from Custom SSH password changers due to a security concern in the logging messages. To resolve this issue, the original security messaging was addressed, and then debug logging was reinstated.
- Fixed an issue where heartbeat failed when changing a AS/400 Mainframe secret due to versioning requirements. When setting up an AS400 Password Changer for V7R1 or V7R2 systems, the a timeout response error was thrown from the password changing that failed to manage the WS3270 utility properly. We updated the supported versioning for the IBM iSeries Password Changer template to resolve this issue.
- Added the option to include subfolders when configuring event subscriptions. Users can now optionally apply event subscriptions to subfolders, previously folder subscriptions were limited to individual folders.
- Enhanced performance of the Event Subscriptions page. Page performance on the event subscription page in Secret Server was increased by rearranging the logic and ordering of the event subscription processes. Secret Server
- Increased the default LDAP processing throttling limit on Secret Server distributed engines (SSDE) from 100 to 1000. SSDE performance was slowed down by the low throttling threshold. Increasing the limit allows faster performance for engine tasks like heartbeat or password changing.
- Enhanced audit logging messages when viewing and displaying secrets and passwords. Unmasking a password or viewing a password's history is only logged in Secret Server every five minutes when the same user performs the same action on a secret within a short period of time. Prior to this update, the description of behavior on the Audit View page did not include the action for "Password Displayed."
- Two-Factor Authentication now supports the User Principal Name (UPN) as a default for usernames when logging into SS. Configure this setting in Admin > Configuration > Login > RADIUS Default Username when

RADIUS is enabled.

- Renamed the "Google Authenticator" dropdown option for multi-factor authentication to "TOTP Authenticator" (Time-based One-time Password algorithm) for accuracy.
- Updated SSDE to use REST instead of Windows Communication Foundation (WCF) when using HTTP to contact Secret Server. WCF is no longer a prerequisite for installing SSDE.
- Added log message buffering. By default, log messages are not buffered and are written to logs immediately. For very active systems, log message buffering can increase performance. In the web-log4net.config file, the Delinea.BufferingForwardingAppender parameter should be uncommented and then IIS reset. Buffer size can be configured using the bufferSize parameter in that configuration file.
- Increased the number of default index separators to include the following characters (separators can be configured under Admin > Search Indexer): ?, !, @, #, (,), [,], {, }, ', ", -
- Significantly improved performance of UI operations through SSDE, such as AD login through SSDE and site validation.
- Blocking calls on AzureServiceBus in cloud no longer create temporary queues, instead they use a persistent queue with a unique session within the queue. This substantially improves the performance of blocking calls.

Bug Fixes

- Fixed a bug where manually added fields in custom templates would not display in the table view. Prior to this fix, when creating a custom template with the "expose for display" option selected, the field would not display in the table as a new column.
- Corrected error messaging when deleting a dependency on secrets with large numbers of dependencies. When a secret had 1000+ dependencies, deleting one of them resulted in a "Failed to Execute" error prior to this fix.
- Fixed error reporting when a user enters a group name that exceeds the maximum character length.
- Resolved an issue with Amazon secret heartbeats and rotations because of a change in Amazon login page layout and flow. To create a new Amazon Web Services (AWS) secret, create a new web password secret, then select Remote Password Changing tab > Password Type (Amazon, Google, and Salesforce options).
- Fixed an issue where a RADIUS Authentication timeout would block other RADIUS requests from authenticating to Secret Server, causing login delays for RADIUS users. Prior to this update there was a lock in the RadiusUSRequest.GetResponse() that only allowed one connection at a time. This fixes an issue when making more than one UDP connection to the same client port.
- Removed unnecessary logging on the OperationCanceledException in the system log.
- Fixed a bug where SSH key-rotation commands were not properly authenticating. The "Verify" command for password changing was ignored if no "Post Change" command existed in a Custom SSH Key Rotation template, resulting in being able to use the same key to connect for the verify command and the password change in certain cases.
- Fixed a broken documentation link on the Server Nodes page.
- Fixed a bug where duplication errors occurred if scheduling a report with the same name as another, already disabled, report in Custom Reports. The workaround for this issue was to undelete the original report and rename it before deleting it, but users are now able to delete and create reports with the same name, if the duplicated reports are not both active.

- Fixed an issue that prevented users from creating or editing an SSH command menu on individual secrets.
- Fixed an issue where the "Allowed" and "Available" Secret Templates columns did not populate on the "Create New Folder" page when restricting secret templates.
- Fixed a bug where editing checkout hooks saved changes instead of allowing users to cancel out of the edit page. After clicking cancel on the checkout hooks edit page, the updated data was not saved to the database but was updated on the page behind the UI modal.
- Edited PUT /secrets/{id}/fields/{slug} parameters in API documentation to use Secret ID instead of the secret name.
- Edited /secrets/{id}/check-in REST API call in API documentation. In this script, "Force Checkin" could bypass the access responses, then successfully check-in the secret, before attempting to return the access responses it previously skipped. This failed.
- Fixed an issue where ExpiredSecretMonitor stopped running in certain conditions due to a session recording call. Users were experiencing several days in between this issue. The SessionRecordingBlobWriter registered an exception error that stopped background threads from processing. This SQL call was irrelevant, and its elimination allowed the ExpiredSecretMonitor to update every minute as normal.
- Fixed an issue where "Language Resource Not Found" errors were thrown on the Themes page under the Advanced section. This issue resulted from missing resource strings on the page.
- Fixed an issue where the SSMS launcher did not send the correct password if a caret symbol (^) was in the password. This issue was specific to SQL Server Management Studio (SSMS.exe). The fix for this involved updating the SSMS process launcher configuration to allow for an optional escape character to be added, now configurable in the advanced section when setting up a launcher for a SQL Server Account secret.
- Fixed a bug where failed password changes on custom SSH secrets would not stop processing when CheckContains command failed. This issue stemmed from the CheckContains command script in the Custom SSH password changer.
- Fixed an issue where items could be imported into the root personal folder. Prior to this fix, administrators were able to migrate secrets into the Personal Folder (root) using the upload XML tool, which allowed all users to see the imported items.
- Fixed an issue where a language setting caused errors. Some language settings could cause an IIS crash with HTTP Error 503: The Service is Unavailable.
- Edited tooltip wording for the advanced "Auto Change Schedule Interval" setting.
- Fixed an issue where SSH key rotations were not properly rotating and then deleting the old SSH key at endpoint when the SSH Custom password changer was configured in a specific way. Key rotations for Linux SSH keys failed to cleanup old SSH keys on target machines after a key rotation occurred due to a missing command in the success script.
- Fixed an issue where users were unable to delete reports. Deleting and undeleting Reports threw an exception error due to a database ReportCategoryId mismatch.
- Fixed an issue where the API call "GET /secret-templates" did not support the inactive filter ("filter.includeinactive").
- Updated API Rest documentation to not label a dictionary object (enumerated KeyValuePair) as an object[].

- Fixed an issue where scheduling reports failed if the time zone was set to time zones ahead of UTC. Prior to this fix, any time zone that was UTC+x failed to properly populate ScheduleCustomReportEdit.aspx "Recurrence Scheduled Start Time."
- Fixed an issue with the API Folder Create method where folders created through the API would error when trying to access permissions. After creating a new folder via the REST API, a Get command for the new folder would return null permissions prior to this fix.
- Fixed a bug where Basic Users were not able to use the "Create New" secrets template when Active Directory template permissions were revoked. In Basic User mode the "Create New" secret templates dropdown list incorrectly started with `<active directory="">`, causing permissions errors when the Active Directory template was revoked for Basic users. The dropdown list now starts with `<select>` in all user modes.
- Fixed a bug where searching for a subfolder would not return results when the user did not have permissions to view its parent folder.
- Fixed a bug where the bulk operation mode did not move multiple secrets to a folder. This issue specifically involved the search feature on the bulk operation dropdown when filtering for the folder location. Users were required to manually select the folder rather than being able to find it through search.
- Fixed issue where "Require Approval Access" on a secret policy did not follow default settings to not enforce the policy.
- Fixed a bug where a user's display name could be left null when creating new users.
- Fixed an issue where discovery computer scans using a "machine only" resolution type resulted in an exception error rather than completing the scan. A null exception was thrown when scanning for machine names rather than using the "Use Fully qualified name (recommended)" setting in discovery.
- Enhanced error messaging when running dependency scans in discovery. Error message fixes included:
  - If all scanners worked but did not find anything: "No Dependencies."
  - If one scanner fails, then it shows a failure message for the scanner.
  - If multiple scanners fail, it only has room to show one failure.
  - If a scan fails but no failure message exists, it shows "Unknown Error."
- Updated two collation settings in session recording tables to allow case insensitivity on table names and prevent collation errors when the SQL server collation mismatches.
- Fixed a bug where the "Only change password when Secret is expired" check box would not save if Auto-Change Schedule was set to None.
- Fixed an issue where SAML authentication check could not be disabled. Prior to this fix, some accounts would receive an 'Invalid Relay State' error when logging in. This had to do with disabling the "SAML authn context" in Secret Server.
- Enhanced error messaging when heartbeat is not assigned to a PowerShell Script password changer.
- Fixed a bug where log file exports were not downloading from the Manage Secret Access Request page.
- Fixed a bug where the "Use Custom Window Size" launcher setting was not implementing the correct resolution. Prior to this fix, when attempting to launch at a smaller 1024 × 768 resolution, a session launched instead as full screen.

- Fixed an issue where if the Response Bus Site Connector disconnected, it prevented the web site from loading. Updates were made to BackgroundScheduler, BackgroundWorker, and the EngineWorker. If a role cannot connect to its response bus an error will now be logged and the site will still load.
- Fixed a bug where heartbeat on local administrator accounts on Windows Server 2012 pre-R2 was not compatible with PowerShell v3.0.
- Fixed a bug where API calls against restricted secrets threw an object reference error. This fix allows accessing "require comment," checkout, or other restricted secrets through the API. These secrets were returning errors due to a null reference linked to a "ViewTracker" secret attribute.
- Fixed an issue where web password secrets caused Bookmarklet JQuery Exceptions when the web password filler was used in the IE11 browser.
- Fixed a bug where EnableFrameBlocking was not respected on certain pages in Secret Server. Middleware was added to send X-Frame-Options / SameOrigin header for everything but the bookmarklet if not installed, no one was logged in, or FrameBlocking was enabled. MainLayout also changed for MVC pages so that they will render with the frame blocking script.
- Fixed a bug where email configurations were not properly saving applied settings. Prior to this fix, changes to Email configuration page did not properly update the database.
- Fixed workflow issue where users were directed to the Setup Home page after saving changes on the Email configuration page.
- Fixed an issue where AD users were not receiving approval emails after an access request for a secret in some environments.
- Fixed a bug which caused audit tables to display date and time in UTC rather than the configured time zone.
- Fixed a bug that could cause a "Failed to load history" error when viewing the history data for a secret.
- Fixed a timeout on secret audit when there are very many audit entries. Audits with very large result sets only display the top 1000 entries.
- Resolved an error reporting issue with Unix Account (SSH) and Unix Account Custom (SSH) account types so that connection failures and login failures are correctly reported. Previously, Unix Account (SSH) would report login failure as UnknownError and Unix Account Custom (SSH) would report login failure as UnableToConnect.
- Updated the REST API call [Get folder-permissions] parameter for [filter.userId] to return all permission information specific to a user.
- Fixed an issue where discovery imported group managed service accounts (gMSAs) as new computers and subsequently attempted to scan them for accounts.
- Fixed a bug where a Java script "page unresponsive" error was caused after moving multiple dependencies into multiple groups; then editing multiple dependencies.
- Fixed an issue where out-of-the box dependency scanners (Scheduled Task, Application Pool, Service Account) were not correctly copying as the same specified scanner's type.
- Fixed an issue where "Optimize Start Menu Cache Files" Windows 10 tasks were logging as dependencies during discovery for some customers, resulting in excessive logging that slowed cloud performance.
- Fixed an issue where converting a Unix Account (SSH Key Rotation) secret template did not allow the target field for "File" types to populate.

- Fixed a bug where incorrect domain names passed on the logon window for RDP Launcher sessions.

Cloud Release Date 12/22/2018

Bug Fixes

- Fixed an issue where SSH Keys were not being passed to the launcher.
- Fixed an issue where the Captcha script was prevented by Content Security Policy on the login page.
- Fixed an issue where processing recorded session failed with Null Reference Exception.
- Fixed a bug where data could be truncated when saving new user.
- Enhanced login workflow in Delinea One

Cloud Release Date 11/03/2018

Enhancements

User Interface

- The new and improved redesign of Secret Server is here. Click the banner in Secret Server to enable for use. For feedback opportunities please inquire at ux@thycotic.com.
- Included a "24 hours" option to the expiration Quick Picker for the Secret Access Request form.

Time Zone Enhancements

Improved timezone handling in Remote Password Changing to correctly account for daylight saving time.

APIs

- Added new service to API to Cancel Password Change.
- Updated the UserCreatedArgs to include AD Guid to account for possible duplicate users in Active Directory through the API.

Launchers

Added backwards & forwards compatibility to the Secret Server Launcher Protocol Handler.

Business Users

SSC now has licensing to support Business users.

- The minimum RPC Retry Interval was increased to 5 minutes. Existing customers with RPC Retry Intervals set lower than 15 minutes will notice an increase upon upgrade to automatically run every 15 minutes, but this setting can be manually configured by 5-minute increments.
- RPC Retries are now limited to 500 consecutive attempts.
- The minimum Heartbeat interval was increased to 60 minutes.
- Added the ability to use a $$Pause Command for the custom SSH Password Changer to allow the option to prevent run commands executing immediately after login.

Security Fixes

- Fixed an issue with Request Access when set to be required for Owners and Approvers where an Approver could approve their own request through JavaScript manipulation. Note an email detailing the approval would have been sent to all other approvers.
- Fixed a Security Vulnerability that allowed Cross-Site Scripting.

Bug Fixes

- Fixed a bug where users with the "Administer Configuration" Role Permission could see but not access option settings under Admin > Configuration.
- Fixed a bug where the Application Pool scanner and IIS tester did not properly initialize the ManagementScope for WMI calls to Local Machines.
- Fixed a bug where only one Discovery Scanner worked when Active Directory had more than one local account scanner setup.
- Fixed a bug where valid REST/SOAP API tokens did not permit login access.
- Fixed an issue with the Mac Launcher.
- Fixed an issue that allowed personal folders to be moved to the top level of the folder tree.
- Fixed an issue where after adding and saving an Identity Provider in advanced settings (SAML), the user observed an error stating 'Object reference not set to an instance of an object.'
- Fixed a bug where a user was unable to delete an approved user from the Edit Secret Policy page.
- Fixed a bug where there were inconsistencies in the time zone handling.
- Fixed a bug where there was an error message of 'Connection Failed' during Custom Unix (SSH) Remote Password Changing but the password changed successfully.
- Fixed a configuration issue with the SSH Key Rotation Privileged Account Password Changer that is used by the Unix Account (Privileged Account SSH Key Rotation) template and caused RPC to fail with an error saying 'Associated Secret is required.'
- Fixed a bug where new users assigned to Google/OATH 2FA could not login if the Remember Me time interval elapsed.
- Fixed a bug where Scan Item on Password Type would not save upon changing scan templates.
- Fixed an issue where Web Password secrets caused Bookmarklet JQuery Exceptions when the Web Password Filler was used in the IE11 browser.
- Fixed a bug where EnableFrameBlocking was not respected on certain pages in Secret Server.
- Fixed an issue where high volume of approvers were not all receiving approval emails after an Access Request for a Secret.
- Fixed an issue where users were able to disable Site Connectors assigned to the Response Bus.
- Added messaging to notify when a user enters a group name that exceeds the max character length.
- Fixed workflow issue where users were directed to Setup page after saving changes on the Email configuration page.

- Fixed a bug where changes to Email configuration page did not update the SQL database.
- Fixed a bug where SSH Key rotation commands were not properly authenticating.
- Fixed an issue where a Radius Authentication timeout would block other Radius requests from authenticating to Secret Server, causing login delays for Radius users.
- Enhanced a permission-related error message within Secret Server to be 'You do not have Owner permission on [Secret Name].'
- Fixed a bug where log file exports did not download from the Manage Secret Access Request page.
- Fixed a bug where the "Maximum Number of Entries" field was disabled after changing it from "Unlimited" on the System Log page.
- Fixed an issue where a WinRM warning message displayed even when WinRM is running and correctly configured on a server if authentication account did not have access to running services.
- Performance enhancement fixes to Distributed Engine.

Cloud Release Date 8/17/2018

Enhancements

- Customers are now able to configure the time when session recordings are moved from the database to disk.

Bug Fixes

- Fixed an issue where new Delinea One customers were unable to access their personal folders after provisioned
- Fixed issue where after 10.5 upgrade, session recording could not be configured to use a file share
- Fixed an issue where there was an intermittent first login fail on fresh install
- Fixed issue where selecting "Now" for a password change in Autochange Schedule could set the incorrect time in situations where the user's computer is in a different time zone than the Secret Server host.
- Fixed issue where dependency scanning would miss Scheduled Task dependencies in situations where two domains with the same name (ex: example.com andexample.net) were used.
- Fixed an issue where Distributed Engines were inconsistently failing after upgrade.
- Fixed issue where the password for an account would be changed multiple times in a row if the "Only Change password when Secret is expired" box was selected in Autochange Schedule.
- Fixed issue where bulk operations could cause database deadlocks.
- Fixed issue where Distributed Engine callbacks could cause a race condition.
- Added a warning to the schedule page that specified the time will be saved in Secret Server configuration time

Cloud Release Date 8/4/2018

Enhancements

Login Flow

- New Login authentication flow in Delinea One is now available
- SAML
- Added ability to a new SAML configuration page so SSO providers can be configured without modifying the saml.config file.

Cloud Key Management

AWS Key Management is now supported

Session Recording

- Cloud support for Session recording
- MP4 conversion through Azure Media Services
- Improved Robustness and speed when generating videos

SIEM

Added new Syslog event messages for SIEM integration and enhanced log messaging.

UI Updates

- Subfolders can now be created within a user's Personal Folder.
- The login screen domain dropdown menu can now be disabled for customers that wish to hide the list of domains.
- Added auditing for the following configuration changes: Character Sets, Password Requirements, Event Subscriptions, Role names, Backups, Custom Password Changers, Licenses, and Database Connections.
- Added export functionality for Heartbeat logs, Remote Password Changing logs, Discovery logs, and Computer Scan logs.
- Failed password changes now display an error within the Secret View UI and a link to a password changing errors KB article.
- If using multiple devices per user in Duo Security for Two-Factor logins, Secret Server will now show the Device Name set in the Duo admin portal next to each device.

Discovery

- Added Discovery settings to scan for open ports on target machines, connect to specific ports, and set a default timeout for port scanning.
- Added a Discovery scanner setting for excluding services, tasks, or application pools by name.
- Added new Diagnostic logs to address duplicate Discovery Scan items.

Password Changing

- Auto Change Schedules can now be configured so that they will trigger a password change even if the Secret is not expired.
- Added the ability to rotate SSH Keys with no passphrase required.

Reporting

- Added Session Revocations to the User Audit report.
- Added IP addresses to the login failure report.
- Added new chart options for custom SQL reports.

Other Notable Enhancements

- Enhanced Secret Server's ability to process larger message sizes. Secret Server'
- Distributed Engines now send operation results back to Secret Server through the Site Connector instead of sending them via website.
- Ticket System Integration can now be configured to work over Distributed Engine.
- Improved System Log Searching in active environments.

Security Enhancements

- Added a configuration setting to change the default role that a new user receives.
- Added a configuration setting to limit the file type and file size of files that are uploaded and attached to Secrets.
- Added the ability to audit certificate verification errors for Active Directory calls over LDAPS and syslog connections using Secure TCP.
- Added the ability to send a client certificate with Active Directory calls over LDAPS or syslog connection using Secure TCP.

Bug Fixes

- Fixed issue where Secret Policy changes would not apply to all Secrets.
- Fixed issue where Service Account Discovery could timeout and flag Secret Dependencies for removal.
- Fixed issue where Two-Factor could prevent a "Login Failed" audit on the user; added new logging details in the Audit User Log if errors do occur from Two-Factor authentication.
- Fixed issue where excluding OUs from Discovery scans prevented computers from being deleted when they were removed from AD.
- Fixed condition in certain environments where the auto-change Secrets were not changing properly. Improved performance of the Discovery Stored Procedure for specific OUs scanning to avoid timeout in large environments.
- Fixed a logging exception in Monitor Logging.
- Resolved a permission error in certain environments that occurred during Local Account Discovery Scans.

- Fixed issue where integrated Windows login requests were building up in the tbOauthExpiration table.
- Fixed issue where columns could not be sorted in Discovery Network View.
- Fixed bug where an email config port change was logging the new port as the old port.
- Fixed an issue with SQL Replication where indexes on indexed views were not replicated.
- Fixed issue where a DependencyResolutionException could occur on the Login page and prevent use of site until an IIS reset was performed.
- Fixed issue with SSH password rotation/Heartbeat connections that were reporting "Unexpectedly inactive."
- Fixed issue where the Secret Server Clipboard Utility could not be installed with Chrome 67.
- Fixed issue where the dashboard Add Content dropdown displayed below the Secrets table.
- Fixed null reference bug that occurred when autocomplete textboxes were used in lieu of a dropdown for the Group/User selector.
- Fixed bug in Password Changing where a large number of Secrets targeting the same resource for certain password changers could prevent processing.
- Fixed issue where root folders could be created through the CSV import process while that user did not have the Create Root Folders role permission.
- Fixed an issue that could cause occasional black flickering to appear in Session Recording videos.
- Fixed issue where users could be logged out of Secret Server due to inactivity while actively browsing certain pages.
- Fixed issue where new Secrets displayed the option to save to a user's Personal Folder even if Personal Folders were disabled.
- Fixed issue where Discovery local account scans were parsing unnecessary data and taking more time than necessary in large Active Directory domains
- Fixed issue where changes to Users would not save when extremely large numbers of AD groups were being synchronized.
- Fixed issue with custom PowerShell Ticket System integrations where entering a ticket number to view a Secret would produce an error.
- Fixed issue where creating a new Event Subscription would fail when specifying a user or group.
- Fixed issue where the REST API would not correctly implement the "Require Two-Factor for Web Services" configuration option.
- Fixed issue where Heartbeat could be stuck in Pending status.
- Fixed issue where creating new Folders would fail when there were no other Folders.
- Fixed condition in certain environments where the auto-change Secrets were not changing properly.
- Fixed issue where Dashboard searching was slow in environments with large numbers of Secrets.
- Fixed an issue where after adding and saving an identity provider in advance settings (SAML), the user observed an error stating 'Object reference not set to an instance of an object.'

Security Fixes

- Fixed issue with Unlimited Admin permissions and managing Groups.
- Deprecated TLS 1.0 in Security Hardening Check.
- Fixed issue with script names.
- Secret data is now authenticated with HMAC-SHA-256.
- SOAP API updated to use the same token generation as the REST API.
- Previously issued SOAP or REST tokens are no longer valid. If you have a saved token in your scripts or code, you must get a new token after updating Secret Server.