Centrify for Web Applications

Authentication Guide for Apache Servers

October 2018 (release 18.11)

Centrify Corporation

Legal Notice

This document and the software described in this document are furnished under and are subject to the terms of a license agreement or a non-disclosure agreement. Except as expressly set forth in such license agreement or non-disclosure agreement, Centrify Corporation provides this document and the software described in this document "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Some states do not allow disclaimers of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of Centrify Corporation, except as otherwise permitted by law. Except as expressly set forth in such license agreement or non-disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of Centrify Corporation. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. Centrify Corporation may make improvements in or changes to the software described in this document at any time.

U.S. Government Restricted Rights: If the software and documentation are being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), in accordance with 48 C.F.R. 227.7202-4 (for Department of Defense (DOD) acquisitions) and 48 C.F.R. 2.101 and 12.212 (for non-DOD acquisitions), the government's rights in the software and documentation, including its rights to use, modify, reproduce, release, perform, display or disclose the software or documentation, will be subject in all respects to the commercial license rights and restrictions provided in the license agreement.

Centrify, DirectControl, DirectAuthorize, DirectAudit, DirectSecure, DirectControl Express, Centrify for Mobile, Centrify for SaaS, DirectManage, Centrify Express, DirectManage Express, Centrify Suite, Centrify User Suite, Centrify Identity Service, Centrify Privilege Service and Centrify Server Suite are registered trademarks of Centrify Corporation in the United States and other countries. Microsoft, Active Directory, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Centrify software is protected by U.S. Patents 7,591,005; 8,024,360; 8,321,523; 9,015,103; 9,112,846; 9,197,670; 9,442,962 and 9,378,391.

The names of any other companies and products mentioned in this document may be the trademarks or registered trademarks of their respective owners. Unless otherwise noted, all of the names used as examples of companies, organizations, domain names, people and events herein are fictitious. No association with any real company, organization, domain name, person, or event is intended or should be inferred.

Contents

About this guide 3

Intended audience 3

Using this guide 3

Conventions used in this guide 4

Finding more information 4

Contacting Centrify 5

Getting additional support 5

Using Centrify and Active Directory for authentication 6

How the Centrify agent supports authentication 6

Using Centrify with Apache servers 7

Single sign-on through Active Directory Federation Service 8

Using Centrify with Active Directory Federation Services 9

Installing Centrify for Apache 10

Preparing your current environment 10

Installing Centrify for Apache 11

Adding Centrify for Apache software to the Apache server 19

Testing authentication using the sample applications 22

Upgrading Centrify for Web Applications 23

Configuring the Apache server for authentication 24

Understanding the supported authentication services 24

Additional services available 26

Modifying Apache directives for authentication 27

Verifying authentication on your own 36

Configuring silent authentication 38

Understanding Internet Explorer security zones 38

Configuring Firefox to allow silent authentication 40

Configuring an Apache HTTP server cluster 42

Centrify software requirements 42

Configure a clustered environment with a reverse proxy 43

Configure a clustered environment with a load balancer 44

About this guide

The *Authentication Guide for Apache Servers* explains how to install and configure the Centrify web application support package for Apache servers. This package enables you to provide Active Directory authentication for web browser clients on Apache servers.

The authentication services described in this guide require you to install a Centrify agent on the Apache host computer. For information about installing the Centrify agent on a Linux or UNIX computer prior to configuring the Apache server, see the *Planning and Deployment Guide*.

Intended audience

This guide is intended for Apache administrators and application developers who are responsible for managing access to applications running on Apache servers. You should have a working knowledge of your Apache environment and be familiar with performing administrative tasks in that environment.

This guide also assumes you have the Centrify agent installed on your Apache server and access to the full library of documents for *Administrator's Guide for Linux and UNIX*.

Using this guide

This guide describes how to install Centrify libraries on an Apache server, test your configuration with a sample application, and configure other applications to use authentication services through Active Directory. The guide is organized as follows:

- Using Centrify and Active Directory for authentication provides an overview of how Centrify for Apache provides authentication services to Apache servers and applications through Active Directory or Active Directory Federation Services.
- Installing Centrify for Apache describes how to install the Centrify for Apache package and how to add the Centrify for Apache libraries to the Apache server.
- Configuring the Apache server for authentication describes how to configure Apache-based applications to use Centrify for Apache and Active Directory for authentication and authorization services.
- Configuring silent authentication describes how to configure Internet Explorer security zones to allow for silent authentication.
- Configuring an Apache HTTP server cluster describes how to configure the computers when the Apache servers are in a cluster.

Conventions used in this guide

The following conventions are used in this guide:

- `Fixed-width` font is used for sample code, program names, program output, file names, and commands that you type at the command line. When italicized, the fixed-width font is used to indicate a variable. In command line descriptions, square brackets (`[ ]`) indicate optional arguments.
- **Bold** text is used to emphasize commands, buttons, or user interface text, and to introduce new terms.
- Standalone software packages include Centrify version and platform architecture information in the file name.

Finding more information

Centrify provides extensive documentation targeted for specific audiences, functional roles, or topics of interest. If you want to learn more about Centrify and Centrify products and features, start by visiting the Centrify website. From the Centrify website, you can download data sheets and evaluation software, view video demonstrations and technical presentations about Centrify products, and get the latest news about upcoming events and webinars.

For access to documentation for all Centrify products and services, visit the Centrify documentation portal. From the Centrify documentation portal, you can always view or download the most up-to-date version of this guide and all other product documentation.

To get to the documentation portal, go to docs.centrify.com or https://www.centrify.com/support/documentation.

Contacting Centrify

You can contact Centrify by visiting our website, www.centrify.com. On the website, you can find information about Centrify office locations worldwide, email and phone numbers for contacting Centrify sales, and links for following Centrify on social media. If you have questions or comments, we look forward to hearing from you.

Getting additional support

If you have a Centrify account, click Support on the Centrify website to log on and access the Centrify Technical Support Portal. From the support portal, you can to search knowledge base articles, open and view support cases, download software, and access other resources.

To connect with other Centrify users, ask questions, or share information, visit the Centrify Community website to check in on customer forums, read the latest blog posts, view how-to videos, or exchange ideas with members of the community.

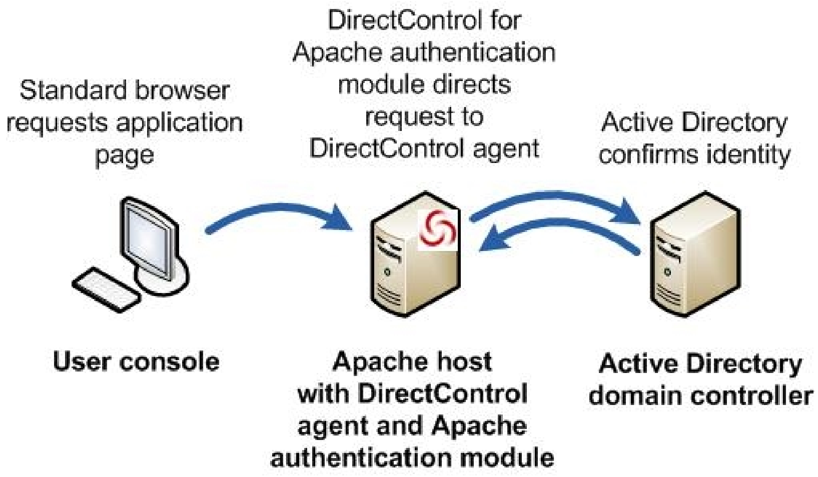
Using Centrify and Active Directory for authentication

This chapter describes how the Centrify for Apache authenticates Apache application users using Active Directory and Active Directory Federation Servers.

How the Centrify agent supports authentication

When the Centrify agent is in place, a Linux or UNIX computer that is joined to an Active Directory domain becomes an Active Directory client for authentication, authorization, policy management, and directory services. The Active Directory environment typically consists of a single Active Directory identity store for all Windows and UNIX users.

Centrify for Web Applications for Apache provides an additional library to extend authentication services for applications and web pages hosted on an Apache server. The following figure provides a simplified view of the communications between the Centrify for Apache authentication module, the Apache server, and Active Directory.


Standard browser requests application page
DirectControl for Apache authentication module directs request to DirectControl agent
Active Directory confirms identity
User console
Apache host with DirectControl agent and Apache authentication module
Active Directory domain controller

The Apache authentication libraries direct standard browser requests for access from the Apache server through to the Centrify agent to Active Directory. The Centrify agent receives the authentication and authorization from Active Directory and returns this information to the Apache server.

Before you can use Centrify for Web Applications for Apache for authentication using Active Directory, you need to:

- Install the Centrify agent on the Apache server Linux or UNIX host.
- Add Active Directory accounts for each user on the domain controller. (The accounts do not need to have the Centrify profile properties, for example UID or GID.)
- Join the Apache server to the Active Directory domain controller.

Using Centrify with Apache servers

The Centrify agent, `adclient`, provides authentication and authorization for basic Linux and UNIX services such as `login` and `telnet`. The modules in Centrify for Web Applications for Apache work in conjunction with `adclient` and the internal service library to provide silent and prompted authentication and authorization when users access Web applications created in Apache environments.

In an Apache server environment, directives are used to configure authentication and authorization for applications. The Centrify-defined directives support the following authentication methods for Web pages, directories, virtual Web sites, and applications on Apache in a standard Active Directory environment:

- Simple and Protected GSS-API Negotiation (SPNEGO): With the SPNEGO authentication method, users who successfully sign into the domain can be silently authenticated to the Web application without entering a user name or password if they use a Web browser that supports SPNEGO tokens. For example, if they use Internet Explorer as their Web browser to access an application, they are authenticated transparently with the user name and password they entered when they initially logged on to their local computer.
- NT LAN Manager (NTLM) authentication for Windows clients: With the NTLM authentication method, users can be authenticated silently or by

specifying a valid Active Directory user name and password when prompted.

- Basic authentication (`BASIC`): With the `BASIC` authentication method, the user is prompted in a browser-generated dialog box to provide a valid user name and password. By default Centrify for Apache is configured to use Active Directory accounts to authenticate the credentials. In addition, you can also enable PAM authentication; this is useful when you want to authenticate the credentials against a local repository; for example, /etc/passwd.

If you are using only Active Directory for authentication, skip to Installing Centrify for Apache (The rest of this chapter is only pertinent to those using Active Directory Federation Services.)

Single sign-on through Active Directory Federation Service

When an organization uses Active Directory, users in any trusted forest can sign on once and be authenticated to resources available throughout the forest. Active Directory Federation Services (AD FS) extends this basic single sign-on capability to Internet-facing applications running on Microsoft Internet Information Service (IIS) computers, enabling customers, partners, and suppliers to securely authenticate their identity when using web applications. Depending on the version of the Active Directory Federation Service you use, you can manage access and authentication through the configuration of account and resource servers (AD FS 1.0) or by configuring “relying party” and “claims provider” trusts and claim rule sets (AD FS 2.0).

Centrify for Apache allows you to extend this single sign-on service to web applications running on Apache servers and on Linux or UNIX computers. The Centrify for Apache package includes a separate set of files that enables authentication and authorization through AD FS. The Centrify service is comparable to the Active Directory Federation Services Web SSO agent. You can configure Centrify agent on a server-by-server basis to use either Active Directory or AD FS for authentication.

When you extract the Centrify for Apache package, the libraries required to work with the standard Active Directory environment and the files required to work with the Active Directory Federation Services environment are both available. You identify the environment you want by selecting the authentication module you want to use.

If your web browser is using Active Directory Federation Services to authenticate users, follow the instructions to install the Centrify software for both Active Directory and Active Directory Federation Services authentication. Configure the server for Active Directory authentication first and use the sample applications to confirm that everything was installed properly. After you have verified authentication using the Active Directory sample applications, go to the *Active Directory Federation Services Configuration Guide* to configure the federated services to run the ADFS-related sample applications.

Using Centrify with Active Directory Federation Services

Centrify for Apache in the Active Directory Federation Services environment authenticates users through interactions between the Account and Resource Federation Servers (AD FS 1.0) or relying party and claims provider trusts (AD FS 2.0). Centrify for Apache supports both claims-aware applications and traditional applications.

- For claims-aware applications, Centrify for Apache validates and passes along any verified claims from the Web browser client to the application. Because the application has been designed to understand how to interpret the claims presented in the security token, the application itself decides on the level of service to provide to the client based on these claims presented.
- For traditional applications that do not take advantage of the AD FS claims directly, Centrify for Apache provides custom Apache directives to control access to the application. For example, a page can be configured to require a specific group claim.

If you are using Active Directory Federation Services proceed to Installing Centrify for Apache to install the modules and test applications to support Active Directory AND Active Directory Federation Services. Confirm proper installation with the test applications and then go to the book *Active Directory Federation Services Configuration Guide* to complete the configuration.

Installing Centrify for Apache

This chapter describes how to prepare for and install the Centrify for Web Applications package, including the sample applications.

Preparing your current environment

Before you install and configure Centrify for Apache package, you should check the local host environment to confirm the following:

- If you are using Active Directory or Active Directory Federation Services (AD FS) you have installed and configured for your Windows environment and the target users have accounts on the Active Directory domain controller or AD FS identity store.
- If you want to authenticate UNIX users against local accounts in addition to, or instead of, against Active Directory, use a Pluggable Authentication Module (PAM). (You specify the service when you configure the Apache directives.)
- You have installed Apache server 2.0 or 2.2, know the path to the Apache server configuration files and know how to start and stop the server. You should also verify that the Apache server is configured to support dynamically loaded objects by running the `httpd -l` command—or `apache2 - l` on some systems—and verifying `mod_so.c` is listed for the server.

  Note Centrify for Web Applications supports Apache 2.0 when it is compiled with either the `prefork` or `worker` Multi-Processing-Module.

- You have checked the supported operating environments and system requirements in the *Centrify for Web Applications Release Notes* and verified that the Centrify for Web Applications module can be installed in the local operating environment.

- If you plan to use Active Directory Federation Services for Apache-based applications, the Apache server must run with Secure Socket Layer (SSL) configured.

  Note SSL is not required for Active Directory authentication. However, it increases security because it encrypts the user's credentials when using BASIC authentication.

- You have appropriate permissions to install the package in the host platform.
- You have installed the Centrify agent and the local computer has joined an Active Directory domain. (You can confirm that the Centrify for Web Applications agent is installed and the computer has joined a domain by typing `adinfo` on the command line on the Linux or UNIX computer.)

Installing Centrify for Apache

The Centrify software plugs into the Apache server as a loadable authentication module. Using the module requires some editing of Apache configuration files, but you do not need to recompile or relink Apache to incorporate the package software.

The Centrify authentication module is provided in a platform-specific package and installed using the platform's native installation mechanism. In the following procedures, use the instructions corresponding to your platform.

Use the following steps to install the authentication module on your Apache server.

If you are installing the Centrify software on Apache servers in a cluster, see Installing Centrify for Apache for additional instructions.

1. Log on to the Linux or UNIX computer using an account with root privileges or switch to the `root` user.
2. If you are loading the files from a CD mount the `cdrom` device using the appropriate command for the local computer's operating environment.

Note When you auto-mount the Centrify for Web Applications CD on HP-UX, file names are displayed in the short name (8.3) format. To see the full name mount the CD manually using the `-o rr` (rockridge extensions) flag.

If you have copied the downloaded package or copied the file to another location verify the location and go on to the next step.

3. Change to the directory on the CD or to the directory where you have copied or downloaded the package.
4. Copy the package corresponding to your host operating system and processor type to a directory on the local system where you have read and write access.

   The following table lists the platform options and the associated processor:

Platform	File name host label	Processor type
AIX	aix5.x.tgz	NA
Debian/Ubuntu	deb5--i386.tgz	32-bit Intel architecture
	deb5-x86_64.tgz	64-bit Intel architecture
HP UX	hp11.11.tgz	PA RISC
	hp11.23-pa.tgz	PA RISC
	hp11.23-ia64.tgz	Itanium 64-bit
Red Hat Enterprise Linux	rhel3-i386.tgz	32-bit Intel architecture
	rhel3-x86_64.tgz	64-bit Intel architecture
Solaris	sol8.tgz	SPARC
	sol9-x86.tgz	32-bit Intel architecture
SuSE	suse8-i386.tgz	32-bit Intel architecture
	suse8-x86-64.tgz	64-bit Intel architecture

Note Host labels can change over time. See the release notes for the most current information.

5. Un-zip and -tar the package using the native commands. Then run the installation command corresponding to the target computer platform. The following table shows the default commands for installing the package in different operating environments. (You are not required, however, to use these commands. Use the commands with which you

are familiar.)

To install on	Do this

Red Hat Enterprise Linux	Once you have extracted the file run the following Red Hat Package Manager (RPM) command:
	`rpm -Uvh centrifydc-apache-`*`package`*`.rpm`
	For example, if you want to install on a Red Hat Enterprise Linux server with a 32-bit processor, you would install the software by typing:
	`rpm -Uvh centrifydc-apache-`*`ver`*`-rhel3-i386.rpm`

SuSE Linux
OpenSuSE Linux
	Once you have extracted the file run the following command:
	`rpm -Uvh centrifydc-apache-`*`package`*`.rpm`
	For example, if you want to install on a SuSE server with a 32-bit processor, you would install the software by typing:
	`rpm -Uvh centrifydc-apache-`*`ver`*`-suse8-i386.rpm`

Debian Linux
Ubuntu Linux
	Once you have extracted the file run the following command:
	`dpkg -i centrifydc-apache-`*`platform`*`.deb`
	For example, if you want to install on a Debian 5 server, you would install the software by typing:
	`dpkg -i centrifydc-apache-`*`ver`*`-deb5-i386.deb`

Solaris	Once you have extracted the file, run the following command: `pkgadd -d CentrifyDC-Apache`

HP-UX	Once you have unzipped the file, run the following command:
	`swinstall -s /`*`path`*`/centrifydc-apache-`*`ver`*`-`*`package`*`.depot`
	where *package* is either hp11.11-pa, hp11.23-pa or hp11.23-ia64.
	For example,
	`swinstall -s /`*`path`*`/centrifydc-apache-`*`ver`*`-hp11.23-ia64.depot`
	installs the package for systems with Itanium processors.
	Note that you must specify the full path to the Centrify for Apache depot file.

AIX	Once you have unzipped the file, create the `.toc` file by running the following command:

To install on	Do this

	`inutoc .`
	Then install the software by running the following command:
	`installp -a -d . CentrifyDC.apache`

The install process unpacks the shared object library for several Apache versions and authentication method (Active Directory and AD FS) and a configuration file that simplifies loading.

The shared object libraries are in the following form:

- Active Directory authentication: `mod_auth_centrifydc_xx`
- AD FS authentication: `mod_adfs_centrifydc_xx`

where xx is the Apache version number 20 (for 2.0), 22 (for 2.2) and 24 (for 2.4).

For most targets the extension is `.so`. The lone exceptions is the shared library for the HP UX on the PA RISC platform which has a `.sl` extension and for AIX which has the extension so.0.

The sample configuration file is also version-dependent. It has the following format:

```
centrifyxx[_64].conf
```

where xx indicates the Apache version. If the file name contains `_64`, it indicates the version you use on platforms with a 64-bit processor.

The following sections list the shared library and configuration file directories for each platform supported and the files that are provided.

- Solaris SPARC
- Solaris x86 systems
- Solaris 64-bit systems
- Linux 32-bit
- Linux 64-bit
- Debian and Ubuntu 32-bit
- Debian and Ubuntu 64-bit
- AIX

- HP UX PA RISC
- HP UX Itanium

Solaris SPARC

On Solaris SPARC computers, files are located in the /usr/share/centrifydc/apache/ directory:

```
lib/sparcv9/mod_adfs_centrifydc_20.so
lib/sparcv9/mod_adfs_centrifydc_22.so
lib/sparcv9/mod_adfs_centrifydc_24.so
lib/sparcv9/mod_auth_centrifydc_20.so
lib/sparcv9/mod_auth_centrifydc_22.so
lib/sparcv9/mod_auth_centrifydc_24.so
samples/conf/centrify.conf
samples/conf/centrify-new.conf
samples/conf/centrify20_64.conf
samples/conf/centrify22_64.conf
samples/conf/centrify24_64.conf
```

Solaris x86 systems

Apache 2.0 servers on Solaris can be built with the large file system option (`lfs`). Use the `*_lfs` version of the files if you have the large file system option. There are no `lfs` files for the Apache 2.2 and 2.4 because `lfs` is built in by default.

```
lib/mod_adfs_centrifydc_20.so
lib/mod_adfs_centrifydc_20_lfs.so
lib/mod_adfs_centrifydc_22.so
lib/mod_adfs_centrifydc_24.so
lib/mod_auth_centrifydc_20.so
lib/mod_auth_centrifydc_20_lfs.so
lib/mod_auth_centrifydc_22.so
lib/mod_auth_centrifydc_24.so
samples/conf/centrify.conf
samples/conf/centrify-new.conf
samples/conf/centrify20.conf
samples/conf/centrify20_lfs.conf
samples/conf/centrify22.conf
samples/conf/centrify24.conf
```

Solaris 64-bit systems

On Solaris 64-bit computers, the following files are located in the /usr/share/centrifydc/apache/ directory:

```
lib/64/mod_adfs_centrifydc_20.so
lib/64/mod_adfs_centrifydc_22.so
lib/64/mod_adfs_centrifydc_24.so
lib/64/mod_auth_centrifydc_20.so
lib/64/mod_auth_centrifydc_22.so
lib/64/mod_auth_centrifydc_24.so
samples/conf/centrify.conf
samples/conf/centrify-new.conf
samples/conf/centrify20_64.conf
samples/conf/centrify22_64.conf
samples/conf/centrify24_64.conf
```

Linux 32-bit

For all versions of Linux except Debian and Ubuntu, the following files are located in the /usr/share/centrifydc/apache/ directory. Apache 2.0 servers on 32-bit Linux can be built with the large file system option (`lfs`). Use the `*_lfs` version of these files if you have the large file system option. There are no lfs files for the Apache 2.2 and 2.4 because `lfs` is built in by default.

```
lib/mod_adfs_centrifydc_20.so
lib/mod_adfs_centrifydc_20_lfs.so
lib/mod_adfs_centrifydc_22.so
lib/mod_adfs_centrifydc_24.so
lib/mod_auth_centrifydc_20.so
lib/mod_auth_centrifydc_20_lfs.so
lib/mod_auth_centrifydc_22.so
lib/mod_auth_centrifydc_24.so
samples/conf/centrify.conf
samples/conf/centrify-new.conf
samples/conf/centrify20.conf
samples/conf/centrify20_lfs.conf
samples/conf/centrify22.conf
samples/conf/centrify24.conf
```

Linux 64-bit

For all versions of Linux except Debian and Ubuntu, the following files are located in the /usr/share/centrifydc/apache/ directory.

```
lib64/mod_adfs_centrifydc_20.so
lib64/mod_adfs_centrifydc_22.so
lib64/mod_adfs_centrifydc_24.so
```

```
lib64/mod_auth_centrifydc_20.so
lib64/mod_auth_centrifydc_22.so
lib64/mod_auth_centrifydc_24.so
samples/conf/centrify.conf
samples/conf/centrify-new.conf
samples/conf/centrify20_64.conf
samples/conf/centrify22_64.conf
samples/conf/centrify24_64.conf
```

Debian and Ubuntu 32-bit

Apache 2.0 servers on 32-bit Debian and Ubuntu Linux can be built with the large file system option (`lfs`). Use the `*_lfs` version of the files if you have the large file system option. There are no `lfs` files for the Apache 2.2 and 2.4 because lfs is built in by default.

```
lib/mod_adfs_centrifydc_20.so
lib/mod_adfs_centrifydc_20_lfs.so
lib/mod_adfs_centrifydc_22.so
lib/mod_adfs_centrifydc_24.so
lib/mod_auth_centrifydc_20.so
lib/mod_auth_centrifydc_20_lfs.so
lib/mod_auth_centrifydc_22.so
lib/mod_auth_centrifydc_24.so
samples/conf/centrify.conf
samples/conf/centrify-new.conf
samples/conf/centrify20.conf
samples/conf/centrify20_lfs.conf
samples/conf/centrify22.conf
samples/conf/centrify24.conf
```

Debian and Ubuntu 64-bit

Apache 2.0 servers on 64-bit Debian and Ubuntu Linux can be built with the large file system option (`lfs`). Use the `*_lfs` version iof the files f you have the large file system option. There are no `lfs` files for the Apache 2.2 and 2.4 because lfs is built in by default.

```
lib32/mod_adfs_centrifydc_20.so
lib32/mod_adfs_centrifydc_20_lfs.so
lib32/mod_adfs_centrifydc_22.so
lib32/mod_adfs_centrifydc_24.so
lib32/mod_auth_centrifydc_20.so
lib32/mod_auth_centrifydc_20_lfs.so
lib32/mod_auth_centrifydc_22.so
lib32/mod_auth_centrifydc_24.so
lib/mod_adfs_centrifydc_20.so
lib/mod_adfs_centrifydc_22.so
lib/mod_adfs_centrifydc_24.so
lib/mod_auth_centrifydc_20.so
```

```
lib/mod_auth_centrifydc_22.so
lib/mod_auth_centrifydc_24.so
samples/conf/centrify.conf
samples/conf/centrify-new.conf
samples/conf/centrify20.conf
samples/conf/centrify22.conf
samples/conf/centrify24.conf
samples/conf/centrify20_64.conf
samples/conf/centrify22_64.conf
samples/conf/centirfy24_64.conf
```

AIX

On IBM AIX computers, the following files are located in the /usr/share/centrifydc/apache/ directory:

```
lib/mod_adfs_centrifydc_20.so.0
lib/mod_adfs_centrifydc_22.so.0
lib/mod_adfs_centrifycd_24.so.0
lib/mod_auth_centrifydc_20.so.0
lib/mod_auth_centrifydc_22.so.0
lib/mod_auth_centrifydc_24.so.0
samples/conf/centrify.conf
samples/conf/centrify-new.conf
samples/conf/centrify20.conf
samples/conf/centrify22.conf
samples/conf/centrify24.conf
```

HP UX PA RISC

On HP-UX RISC computers, the following files are located in the /usr/share/centrifydc/apache/ directory:

```
lib/mod_adfs_centrifydc_20.sl
lib/mod_adfs_centrifydc_22.sl
lib/mod_adfs_centrifydc_24.sl
lib/mod_auth_centrifydc_20.sl
lib/mod_auth_centrifydc_22.sl
lib/mod_auth_centrifydc_24.sl
samples/conf/centrify.conf
samples/conf/centrify.conf
samples/conf/centrify20.conf
samples/conf/centrify22.conf
samples/conf/centrify24.conf
```

HP UX Itanium

Separate versions are provided for 32- and 64-bit versions of HP UX.

```
lib/hpux32/mod_adfs_centrifydc_20.so
lib/hpux32/mod_adfs_centrifydc_22.so
lib/hpux32/mod_adfs_centrifydc_24.so
lib/hpux32/mod_auth_centrifydc_20.so
lib/hpux32/mod_auth_centrifydc_22.so
lib/hpux32/mod_auth_centrifydc_24.so
lib/hpux64/mod_adfs_centrifydc_20.so
lib/hpux64/mod_adfs_centrifydc_22.so
lib/hpux64/mod_adfs_centrifydc_24.so
lib/hpux64/mod_auth_centrifydc_20.so
lib/hpux64/mod_auth_centrifydc_22.so
lib/hpux64/mod_auth_centrifydc_24.so
samples/conf/centrify.conf
samples/conf/centrify-new.conf
samples/conf/centfy20.conf
samples/conf/centrify22.conf
samples/conf/centrify24.conf
samples/conf/centrify20_64.conf
samples/conf/centrify22_64.conf
samples/conf/centrify24_64.conf
```

Adding Centrify for Apache software to the Apache server

In this section you add the Centrify for Apache authentication module and sample application directives load instructions. The sample configuration files load the authentication modules for both Active Directory and Active Directory Federation Services. For testing purposes only, load both modules. After you are done testing with the sample applications, configure the load instructions for your environment.

Use the following instructions to load the Centrify for Apache authentication modules:

1. Verify that the Apache server supports dynamically loaded objects.

   You can perform this check by running either the `./httpd -l` or `./apache2 -l`—depending upon your platform—command and verifying `mod_so.c` has been compiled into the Web server.

   If the server supports dynamically loaded objects, you should see `mod_so.c` in the list of compiled in modules.

   If you are building a new server, specify `--enable-module=so` on the command line before doing the `make` and `make install` of your Apache service. For example, your `configure` command might look like this:

   ```
   ./configure --enable-module=so
   ```

Note that the default Apache source code build does not support dynamically loaded objects. For detailed information about building Apache servers, see the appropriate Apache documentation.

2. Edit the Apache server configuration file `httpd.conf` or `apache2.conf`—depending upon your platform—to include the Centrify for Apache authentication module and sample applications directives.

   The simplest way to load the files is to use the `Include` directive and specify the location of the Centrify for Web Applications sample configuration file; for example,

   ```
   include
   /usr/share/centrifydc/apache/samples/conf/centrifyxx.conf
   ```

   where xx is the Apache version.

   For example:

   - For Apache 2.2 on a 32-bit system:

     ```
     Include /usr/share/centrifydc/apache/samples/conf/
     centrify22.conf
     ```

   - For the Apache 2.4 64-bit version:

     ```
     Include /usr/share/centrifydc/apache/samples/conf/
     centrify24_64.conf
     ```

   The configuration script loads both of the authentication modules (`mod_auth_centrifydc_...` and `mod_adfs_centrifydc...`) and the `centrify.conf` (or `centrify-new.conf` for Apache 2.4) file. Alternatively, you can use the `LoadModule` and `Include` directives and to load the files individually. For example,

   - Add the following line to load the authentication module for Apache 2.2 on a Solaris SPARC-based system.

     ```
     LoadModule centrifydc_auth_module
     /usr/share/centrifydc/apache/lib/sparcv9/
     mod_auth_centrifydc_22.so
     ```

   - Add the following line to load the AD FS authentication module for Apache 2.4 on a 64-bit Linux-based system.

     ```
     LoadModule centrifydc_adfs_module
     /usr/share/centrifydc/apache/lib64/
     mod_adfs_centrifydc_24.so
     ```

   Next, add the following line for the sample application directives:

- For Apache 2.0 and 2.2:

  Include /usr/share/centrifydc/apache/samples/conf/centrify.conf

- For Apache 2.4:

  Include /usr/share/centrifydc/apache/samples/conf/centrify-new.conf

3. **Optional:** Use the following instructions to enable Secure Socket Layer (SSL) support for the Apache server. SSL is required if you are using AD FS but optional if you are using Active Directory (use it if you want to encrypt the user's credentials when using BASIC authentication).

   Configuring the Apache server to use SSL varies depending on the Apache version of Apache. For example, on Apache 2.0, you start SSL using the `apachectl startssl` command; however, in Apache 2.2, you configure SSL using directives in the main server configuration file. (See Modifying Apache directives for authentication for more about the directives.)

   - For Apache 2.0, which includes the `mod_ssl` module, you must enable SSL support; for example, your `configure` command might look like this:

     ```
     ./configure --enable-ssl
     ```

     You can start the Apache 2.0 server with SSL by running the `apachectl startssl` command.

   - For Apache 2.x, you can enable and configure SSL settings in directives in the main Apache server configuration file, `httpd.conf` (or `apache2.conf` on some platforms).Once configured, you can start the Apache server with SSL by running the standard `apachectl start` command.

   You can verify whether you have configured support for SSL by opening a browser and trying to access the default web page using `https://localhost/` or `https://servername/`. You should always perform this test if you intend to use authentication service with Active Directory Federation Services.

   Note In an evaluation or lab environment, you can use a local self-signed certificate for testing purposes. In a production environment, however, you should ensure that the security certificates you accept provide an appropriate level of protection.

4. Restart the Apache server to load the new module. For example, if you have installed Apache in the `/usr/local/apache2` directory:

   ```
   /usr/local/apache2/bin/apachectl restart
   ```

This concludes the installation of the Centrify for Apache authentication module and sample application.

The sample configuration file `centrify.conf` you loaded includes the directives you need to run the Active Directory and AD FS sample applications. You can run the sample applications that use Active Directory right away; see the instruction in Testing authentication using the sample applications which follow immediately below.

However, you cannot run the sample applications that use AD FS for authentication. If you are using AD FS for authentication, run the Active Directory sample applications now and then proceed to the *Active Directory Federation Services Configuration Guide* for the next round of instructions.

Centrify for Apache includes extensions to the standard Apache directives that appear in the Apache `httpd.conf` or `apache2.conf` on some platforms and `.htaccess` files. (The `centrify.conf` file demonstrates the use of some of these directives.) In addition, Centrify for Apache uses the environment variables or HTTP header names to set values for authenticated user information. See Configuring the Apache server for authentication for descriptions of the directives and variables/headers used.

Testing authentication using the sample applications

The sample applications are located in the `/usr/share/centrifydc/apache/samples` directory. These samples allow you to test the behavior when an application and the browser are configured with BASIC, NTLM, or Kerberos authentication enabled. The samples verify that a user with an Active Directory account can log on so they are a good test of your configuration.

If you are unsure if your browser supports BASIC, NTLM or Kerberos authentication, see Configuring silent authentication to see the requirements.

By default, the sample directives in `centrify.conf` allow any user authenticated for Active Directory to log on.

By default, the sample directives in `centrify.conf` allow any user authenticated for Active Directory to log on.

To test authentication using the sample applications:

1. Open a web browser and go to the following URL:

   `http://server/samples`

2. Click each authentication option - basic, ntlm and kerberos - and then click **Authenticate** to test the behavior and verify that your test user is authenticated properly.

   For example, if you select the sample that uses `BASIC` authentication, you are prompted to provide a user name and password.

   If authentication is successful, the web page displayed indicates the authenticated user’s identity and other details about the user and web environment.

To run the sample applications in an environment with Active Directory Federation Services, see the *Active Directory Federation Services Configuration Guide*.

Upgrading Centrify for Web Applications

To upgrade from the previous version of the Centrify for Web Applications simply download install the package as described in this chapter and update your `httpd.conf` or `apache2.conf` on some platforms file with the new authentication module and directives.

Configuring the Apache server for authentication

Centrify for Web Applications for Apache is a module, mod_auth_centrifydc_xx, that plugs into the Apache Web server as a loadable module. Once it is loaded, the following sequence of messages are used to authenticate the user when the browser requests a Web page on the Apache Web server

- mod_auth_centrifydc sends back a request to the client indicating which types of authentication (BASIC, NTLM or Kerberos) are supported.
- The Web browser client then sends credentials to the Apache server to authenticate the user.
- mod_auth_centrifydc then sends a request to the DirectControl agent (the adclient daemon) to authenticate and authorize the client using Active Directory.

This chapter describes the authentication services available and the Centrify for Web Applications extensions to the standard Apache directives..

Understanding the supported authentication services

Centrify for Web Applications for Apache supports authentication in Active Directory using the following services:

- Kerberos authentication
- NTLM authentication
- Basic authentication

Each service provides specific features and has its own configuration requirements.

Kerberos authentication

Kerberos authentication provides secure silent authentication for Web browser clients. The client gets a Kerberos ticket for the Web service, then sends its Kerberos credentials to the Web server. The `mod_auth_centrifydc` module then uses Kerberos algorithms to validate the user's credentials.

To enable Kerberos authentication:

- The Web page, Web directory, virtual Web site, or entire Web site must be configured to be protected with Kerberos authentication.
- The Web browser client must support Kerberos. See Configuring silent authentication to learn how to configure Internet Explorer and Firefox to use Kerberos for silent authentication.
- The Windows user to be authenticated must specify an Active Directory domain account by either logging in using an Active Directory domain account or specifying a fully qualified domain name when prompted.
- The Web server must be joined to a domain in the same forest as the client's Active Directory account.

NTLM authentication

NTLM (NT LAN Manager) authentication is a native Windows authentication protocol developed and supported by Microsoft.When Apache NTLM authentication is enabled, the Web browser client can be authenticated based on this Windows authentication protocol.

For Internet Explorer clients, NTLM provides a silent authentication method that can be used in configurations where Kerberos authentication is not possible. For other Web browsers, such as Firefox, NTLM provides a "challenge/response" mechanism that avoids sending passwords in clear text.

To enable NTLM authentication:

- The Web page, Web directory, virtual Web site, or entire Web site must be configured to be protected with NTLM authentication.
- The client browser must be one that supports NTLM authentication, such as Internet Explorer and Firefox browsers.See Configuring Firefox to

allow silent authentication to learn how to configure Internet Explorer and Firefox to use NTLM for silent authentication.

Basic authentication

Basic authentication is a common form of Web site protection. With Basic authentication, the Web browser client prompts for a user name and password and sends this information in either plain or encrypted text to the Web server. In Centrify for Web Applications Basic authentication the default uses Active Directory for Basic authentication (see the `Enable ...` directives in the table Extensions to Apache Directives).

Note The default configuration authenticates all requests are validated against accounts in the Active Directory domain controller. If you want to support local authentication, use the PAM authentication.

Note Using Basic authentication on Web servers that are not configured to use the Secure Socket Layer (SSL) protocol allows user passwords to be sent across the network unencrypted in plain text. In most cases, therefore, you should configure the Web service to use `https` and the Secure Socket Layer (SSL) protocol if you have applications that use Basic authentication. Configuring SSL for a Web service does not require any modifications to your Centrify for Web Applications configuration.

To enable Basic authentication, the Web page, Web directory, virtual Web site, or entire Web site must be configured to be protected with Basic authentication.

Additional services available

Centrify for Apache for Apache supports the following additional services for authentication and authorization. Each service provides specific features and has its own configuration requirements.

Authentication re-prompting

If a user is unable to access a page because of invalid credentials or an authorization failure, Centrify for Web Applications for Apache gives the browser a chance to supply alternate credentials. Although by default Centrify for Web Applications allows reprompting, you can configure it to disable prompting after a Kerberos validation failure, if needed.

Authorization

Authentication establishes the identity of the client. Once this identity has been securely established, Centrify for Web Applications for Apache authorizes the client based on the client's identity or group membership.

Centrify for Apache uses Apache configuration files to specify which users and groups have access to a Web page, Web directory, virtual Web site, or entire Web site. The users and groups specified in the configuration file consist of Active Directory users and groups that belong to a domain in the same forest as the domain to which the Web server system is joined.

Modifying Apache directives for authentication

Centrify for Apache for Apache authentication and access control is handled through extensions to the standard Apache directives that appear in the Apache `httpd.conf` or `apache2.conf` and `.htaccess` files.

Note On some platforms, `httpd.conf` is `apache2.conf`, instead. Your platform has one or the other, and they serve the same purpose.

Once the Centrify for Apache authentication module is loaded into the Apache server, it enables the following extensions to the Apache directives:

Extensions to Apache Directives

Directive	Settings
`AuthName`	The name of the `domain` (realm) under which Basic authentication is performed. This string is used only by the browser in prompting the user for a user name and password.

Directive	Settings
	If the name you want displayed contains blank spaces, you must use quotes in the directive. For example: `AuthName “Zen Communications”`
`AuthType`	The authorization type must be specified as `CENTRIFYDC`, in all uppercase letters.
`CheckPamFirst`	Set true to authenticate the user using PAM first and then Active Directory. This directive is used only if EnableBasicAuth and EnablePamAuther are both true. The default value, if you do not set this directive, is `false`.
`CheckpwdLoggerName`	Set to the logger name for the program set in CheckpwdPath to use for logging messages about PAM authentication. This directive is used only if EnableBasicAuth and EnablePamAuther are both true. If not set, the default is `com.centrify.dc.apache.checkpwd`
`CheckpwdPath`	Set to the full path to the program to call to authenticate users when EnablePamAuth is true. If not set the default is `/usr/share/centrifydc/apache/bin/checkpwd.`

`CustomAttributes`	Set to a list of LDAP attributes, separated by white space, to fetch for the authenticated user.
	The values of the given user's LDAP attributes (if non-empty) will be set in environment variables (if `SetAuthUserInfo` is set to `env`) or in HTTP headers (if `SetAuthUserInfo` is set to `httpheaders`). The form for environment variables is:
	`CUSTOM_ATTR_attr-name = value`
	The form for HTTP headers is:
	`HTTP_CUSTOM_ATTR_attr-name = value`
	For example, if you enter the following attributes, when `SetAuthUserInfo` is set to `env` (assuming a username of `webuser1`):
	`CustomAttributes   cn displayName samAccountName`

Directive	Settings

	the following environment variables are set:
	`CUSTOM_ATTR_cn = webuser1`
	`CUSTOM_ATTR_displayName = webuser1`
	`CUSTOM_ATTR_sAMAccountName = webuser1`

`EnableBasicAuth`	Set to `true` to enable Basic authentication, `false` otherwise. The default value, if you do not set this directive, is `false`.
`EnableKerberosAuth`	Set to `true` to enable Kerberos authentication, `false` otherwise. The default value, if you do not set this directive, is `false`.
`EnableNtlmAuth`	Set to `true` to enable NTLM authentication, `false` otherwise. The default value, if you do not set this directive, is `false`.
`EnablePamAuth`	Set to `true` to enable basic username and password authentication using PAM, `false` otherwise. The default value, if you do not set this directive, is `false`.
`EnableKerberosReprompt`	Set to `true` to enable reprompting the client with NTLM or Basic authentication after a Kerberos validation failure so the client can authenticate as a different user using NTLM or Basic if the Kerberos ticket is invalid. The directives, `EnableNtlmAuth` and `EnableBasicAuth` must also be set to `true` to enable NTLM and Basic reprompting. Set to `false` to disable the server from letting the client attempt login using a different method (NTLM or Basic) when the Kerberos ticket is invalid. The default, if you do not set this directive, is to reprompt (`true`).
`EnableNtlmReprompt`	Set to `true` to enable reprompting the client with Basic authentication after an NTLM validation failure so the client can authenticate as a different user using Basic. The directive, `EnableBasicAuth` must also be set to

Directive	Settings
	`true` to enable Basic reprompting. Set to `false` to disable the server from letting the client attempt login using a different method (Basic) when NTLM authentication fails. The default, if you do not set this directive, is to reprompt (`true`).
`EnableBasicReprompt`	Set to `true` to enable reprompting the client with Basic authentication again after a Basic validation failure so the client can authenticate as a different user but still using Basic. The directive, `EnableBasicAuth` must also be set to `true` to enable Basic reprompting. Set to `false` to disable the server from letting the client attempt login again. The default, if you do not set this directive, is to reprompt (`true`).
`EnableReAuth`	Set to `true` to enable reprompting the client when authorization fails. Use the directive, `Require`, to specify a list of authorized users or groups. Set to `false` to disable the server from reprompting the client for authorization. The default, if you do not set this directive, is not to reprompt (`false`).
`HttpHeaderPrefix`	Set to `PREFIX` to configure a prefix to be added to the HTTP headers to avoid possible conflicts with other proprietary HTTP headers on the server. This directive is ignored if `SetAuthUserInfo` is not set to `httpheader`.
`IdentityType`	Set to one of the following key words to identify the type of authenticated name to set for `REMOTE_USER`: ▪ `UPN` — Sets `REMOTE_USER` to the authenticated user’s Universal Principal Name (UPN). This is the default if you do not specify an `IdentityType`. ▪ `SAMAccountName` — Sets `REMOTE_USER` to the authenticated user’s `SAMAccountName` (the short name).

Directive	Settings
	■ `CommonName` — Sets `REMOTE_USER` to the authenticated user’s `CN` attribute. ■ `FromInput` — Sets `REMOTE_USER` to the user name as entered by the user in Basic user name and password authentication. For Kerberos and NTLM authentication, `REMOTE_USER` is set to the authenticated user’s UPN. ■ `Custom:`attribute-name — Sets `REMOTE_USER` to the authenticated user's attribute-name Active Directory attribute. For example, `IdentityType Custom:mail` ■ `PAM` — Sets `REMOTE_USER` to the user name as entered by the user in PAM user name and password authentication. If `EnablePamAuth` is true and the user was authenticated by PAM, the `IdentityType` is set to PAM regardless of what is set in `httpd.conf` or `apache2.conf` file or `.htaccess`.
`PamService`	If `EnablePamAuth` is true, you can set this directive to identify the PAM service to use. For example: `/etc/pam.d/passwd` If no service is set, the default is `login`.
`Require` *`option`*	Set to limit which users and group members have access. If no `Require` directive is included, all Active Directory or PAM users have access. The `Require` syntax you use depends upon the Apache version. ■ Apache 2.0 and 2.2 `Require user` userID `Require group` groupID `Require valid-user` ■ Apache 2.4 `Require centrify-dc-user` userID [userID]

Directive	Settings

	`Require centrify-dc-group` groupID [groupID]
	`Require centrify-dc-valid-user`
	Use the UPN to specify the userID. Use a space to separate multiple user names. For example:
	`Require centrify-dc-user ray@zen.com star@zen.com`
	Use the full canonical name to specify the groupID. Use a space to separate multiple group names. If the group name contains a space enclose the full canonical name in double quotation marks. For example:
	Require group "*zen.com*/Users/HR Staff"
	Use `valid-user` to permit access to any authenticated domain user. For example:
	`Require valid-user`
	If you are using PAM authentication, the user or group name must be preceded by the `Pam:` prefix. Directives that start with Pam: are ignored for Active Directory users. For example:
	`Require centrify-dc-user Pam:<unixuser>`
	`Require centrify-dc-group Pam:<unixgroup>`

`SetGroupMembership`	Set true to get all groups that the user is a member of and set them in the REMOTE_GROUPS environment variable or the HTTP_REMOTE_GROUPS header. If not set the default is true. Note: Set to false for faster performance
`ReturnStatusForbidden`	Set to `true` to change the return status to **Forbidden** (error 403) instead of **Unauthorized** (401) on authorization failure or on final authentication failure. If not set the default is `true`.
`UseCache`	Set to `true` use the cache in the `adclient` daemon when checking for user group membership for authorization. If not set, the default is `false`.

You can place these directives in either the `httpd.conf` (or `apache2.conf`) or `.htaccess` file, depending on your needs. For example, if you centrally manage the configuration for different directories in the main configuration file, you can add these directives where needed in a single file and maintain them in a single location.

Alternatively, you can provide these directives in separate `.htaccess` files so that different administrators can set their own directives for the directories they manage without making changes to the main configuration file. If you decide to place the directives in individual `.htaccess` files, however, you must include the `AllowOverride` directive in the `httpd.conf` (or `apache2.conf`) file, and be sure that this directive is set to `All` or, at a minimum, set to allow `AuthConfig` directives.

The following is an example of the Centrify for Apache directives set for a specific directory in the main `httpd.conf` (or `apache2.conf`) file:

```
<Directory “usr/local/apache2/htdocs/sample-dir”>
   AuthType                  CENTRIFYDC
   AuthName                  zen.com
   EnableBasicAuth           true
   EnableKerberosAuth        true
   EnableNtlmAuth            true
   EnableKerberosReprompt    true
   Require                   valid-user
   SetAuthUserInfo           httpheader
</Directory>
```

The following is an example of the Centrify for Apache directives in a sample `.htaccess` file for an Apache 2.4 server:

```
AuthType                  CENTRIFYDC
AuthName                  zen.com
EnableBasicAuth           true
EnableKerberosAuth        true
EnableNtlmAuth            true
EnableKerberosReprompt    true
Require centrify-dc-group zen.com/groups/ApacheGroup
SetAuthUserInfo           httpheader
```

Setting the IdentityType directive

You should note that the IdentityType for the REMOTE_USER must be set for the authenticated user or the authentication will not succeed. In addition, the Centrify agent does not retrieve all Active Directory attributes, by default. If you specify an attribute that is not retrieved and cached by agent, authentication will fail. To guarantee that an attribute is retrieved by the agent, you can add it to the `centrifydc.conf` configuration file with the `adclient.custom.attribute.user` parameter.

For example, to specify mail as an attribute to cache, edit the configuration file and add the following line:

```
adclient.custom.attributes.user: mail
```

After editing the file, restart the adclient process and flush the cache with the following command:

```
/usr/share/centrifydc/bin/centrifydc restart -F
```

Modifying standard Apache directives for NTLM

In general, Centrify for Apache directives work seamlessly with the standard Apache directives which you use to control the configuration and operation of the Apache server. In some versions of Apache, however, the default setting for the `KeepAlive` directive is `Off`. This directive setting prevents NTLM authentication. For example, the default version of the Apache server installed with Red Hat Enterprise Linux is configured with the `KeepAlive Off` directive and, therefore, does not allow NTLM authentication by default. To allow NTLM authentication, you need to modify the main Apache configuration file (`httpd.conf` or `apache2.conf`) or the local `.htaccess` file to change this setting.

To allow NTLM authentication in this case:

1. Open the `httpd.conf` (or `apache2.conf`) file or `.htaccess` in a text editor.
2. Locate the KeepAlive directive and check its current setting. For example:

   ```
   KeepAlive Off
   ```

3. Change the `KeepAlive Off` directive to `KeepAlive On`, if necessary. For example:

   ```
   KeepAlive On
   ```

In addition to this change, you may want to modify Apache `KeepAliveTimeout` directive in the `httpd.conf` (or `apache2.conf`) file. The `KeepAliveTimeout` directive controls how long a connection can remain open without any browser interaction. With NTLM authentication, once a connection is established, the user does not need to be re-authenticated as long as the connection remains open. If you are using Firefox with NTLM authentication, you need to set the value for the KeepAliveTimeout directive to allow enough time for the user to type both his NTLM user name and password.

Setting values for authenticated users

Centrify for Apache for Apache uses the following environment variables or HTTP header names to set values for authenticated user information.

Note Use the SetAuthUserInfo directive to specify whether to set authenticated user information in HTTP headers or in environment variables.

This environment variable or HTTP header name	Is set to
REQUEST_AUTH_METHOD	The types of authentication to enable. The valid types are: - Basic - Kerberos - NTLM - PAM - ADFS (Active Directory Federation Services - not described in this book.) The types are not mutually exclusive so more than one type may be enabled. For example: `REQUEST_AUTH_METHOD=basic, kerberos,Ntlm`
IDENTITY	The Universal Principal Name (UPN) of the authenticated user. For example: `IDENTITY=john.doe@acme.com`
IDENTITY_TYPE	The type of the identity claim provided by the IDENTITY variable. For authenticated user information, the only valid identity type is UPN. For example: `IDENTITY_TYPE=UPN`
REMOTE_USERNAME	The authenticated user's samAccountName from Active Directory. The samAccountName supports

This environment variable or HTTP header name	Is set to
	pre-Windows 2000 logon names. For example: `REMOTE_USERNAME=ACME\john.doe`
`REMOTE_UPN`	The Universal Principal Name (UPN) of the authenticated user. For example: `IDENTITY=john.doe@acme.com`
`CUSTOM_ATTR_attr-name`	The value of the user's LDAP attr-name attribute if non-empty. You can configure the LDAP attributes to fetch for the user by using the CustomAttributes directive.

Verifying authentication on your own

To verify that accounts are authenticated using Active Directory, create a test directory within your Apache server’s root directory with a local copy of the authentication directives you plan to place in the main server configuration file (`httpd.conf` or `apache2.conf`) or in individual access control files (`.htaccess`).

To verify authentication:

1. Confirm that the `AllowOverride` directive in the main server configuration file allows authentication directives to be set. You can temporarily change this setting, if needed, for testing purposes. For example:

   ```
   AllowOverride AuthConfig
   ```

2. Create your test directory and a `.htaccess` file with the directives to use. For the `Require` directive, you can specify an existing Active Directory user or group or use `valid-user`.
3. Open your Web browser and attempt to access the test directory using a valid Active Directory logon name and password.

If authentication is successful, you will be logged on and able to access files in the test directory.

You can view information about every successful and failed authentication or authorization attempt in the Apache `error_log` file under the Apache installation directory. For example, the default location for the file in Apache 2.0 is `/usr/local/apache2/logs/error_log`. Any time a user attempts to access a protected Web page, Web directory, virtual Web site, or Web site, details about the success or failure are recorded in the log file.

Configuring silent authentication

This chapter describes how to modify Internet Explorer and Firefox to allow for silent authentication when you are using `SPNEGO` or `NTLM` authentication in your Apache applications.

Understanding Internet Explorer security zones

For users to be authenticated silently when they use Internet Explorer to access an application on the Web server with Kerberos or NTLM authentication, two conditions must be met:

1. Internet Explorer must have integrated Windows authentication enable - see the instructions below.
2. The Web server must be in the **local intranet** Internet Explorer security zone or explicitly configured as part of the local intranet security zone.

   For Internet Explorer, a server is recognized as part of the local intranet security zone in one of two ways:

   - When the user specifies a URL that is not a fully qualified DNS domain name. For example, if you access an application with a URL such as `http://admin-server/index.html`, Internet Explorer interprets this as a site in the local intranet security zone.
   - When the user specifies a URL with fully qualified name that has been explicitly configured as a local intranet site in Internet Explorer (see instructions below). For example, if you access an application with a URL such as `http://admin-server.mycompany.com/index.html`, Internet Explorer interprets

this as a site that is not part of the local intranet unless the site has been manually added to the local intranet security zone.

Depending on whether users log on to Web applications using a local intranet URL or a fully-qualified path in the URL, silent authentication may require modifying the local intranet security zone in Internet Explorer.

Enable Integrated Windows Authentication

Use the following procedure to enable silent authentication on each computer.

1. Open Internet Explorer and select **Tools > Internet Options**
2. Click the **Advanced** tab.
3. Scroll down to the **Security** settings.
4. Check the **Enable Integrated Windows Authentication** box.
5. Restart IE.

Add Web Server to local intranet security zone

If some users log on to Web applications using a fully-qualified path in the URL, they may need to modify the settings for the local intranet security zone in their Internet Explorer Web browser to enable silent authentication.

To configure the local intranet security zone in Internet Explorer:

1. Open Internet Explorer and select **Tools > Internet Options**
2. Click the **Security** tab.
3. Click the **Local intranet** icon.
4. Click **Sites**.
5. Click **Advanced**.
6. Type the URL for the Web site you want to make part of the local intranet, then click **Add**. You can use wildcards in the site address, for example, `*://*.mycompany.com`. When you are finished adding URLs or URL patterns, click **Close**.

7. Click **OK** to accept the local intranet configuration settings, then click **OK** to close the Internet Options dialog box.

Once you have configured the **Local intranet** security zone in Internet Explorer, you can log on to Web or Java applications through Kerberos or NTLM without being prompted to enter a user name and password.

Configuring Firefox to allow silent authentication

By default, Firefox supports prompted NTLM authentication. To enable silent NTLM authentication, you first need to configure the browser to trust sites.

To enable silent NTLM authentication in Firefox:

1. Open Firefox.
2. Type `about:config` as the target URL.
3. Type `ntlm` in the Filter field.
4. Select and right click `network.automatic-ntlm-auth.trusted-uris and select Modify`.
5. Type a comma-separated list of partner URLs or domain names as string values, then click **OK**. For example, type `http://fire.arcade.com,https://fire.arcade.com`, then click **OK**.

   Note For security reasons, you should be as restrictive as possible in specifying this list.

Although the Mozilla Firefox web browser supports negotiated (SPNEGO) authentication, this support is not enabled by default. To enable silent SPNEGO authentication for the Firefox browser, you first need to configure the browser to trust sites.

To enable silent SPNEGO authentication in Firefox:

1. Open Firefox.
2. Type `about:config` as the target URL.
3. Type `neg` in the Filter field.
4. Select and right click `network.negotiate-auth.delegation-uris` and select **Modify**. Enter a comma-separated list of partner URLs or domain

names as string values, then click **OK**. For example, type `http://fire.arcade.com,https://fire.arcade.com`, then click **OK**.

Note For security reasons, you should be as restrictive as possible in specifying the list of trusted sites.

5. Repeat for `network.negotiate-auth.trusted-uris.`

Configuring an Apache HTTP server cluster

This appendix explains how to set up an Apache HTTP server cluster to use authentication service and Centrify for Apache for user authentication.

Centrify software requirements

When you set up Apache servers in a cluster, each server and, if you are using a reverse proxy the reverse proxy computer as well, must have the following Centrify software installed:

- All Linux- and UNIX-based systems: The DirectControl agent (`adclient`) must be installed. Run `adinfo` on each server to confirm that the agent is installed. (Windows-based servers do not require `adclient`.)
- All Linux-, UNIX-, and Windows-based systems: The Centrify for Apache software must be installed.

Note A load balancer is an exception to this rule. If you are using a load balancer, do not install the DirectControl agent or the DirectControl for Web Applications software on the load balancer.

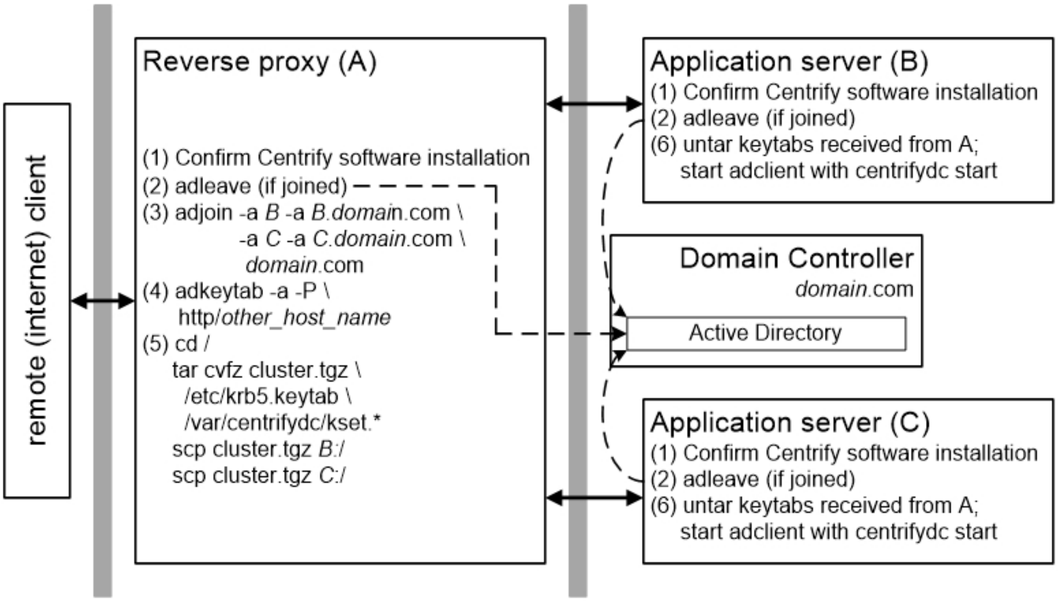
In addition, the Kerberos keytabs for each server must be the same. The following instructions tell you how to copy the keytab across systems.

The next two sections provide sample, step-by-step instructions you can customize for your environment to set up Active Directory authentication in a clustered environment with a reverse proxy and then with a load balancer.

Configure a clustered environment with a reverse proxy

This section assumes that you are installing the Centrify for Apache package in a cluster that has a reverse proxy with multiple servers on the back end.

In the following example, the reverse proxy is running on a machine named A, Apache servers are running on machines named B and C, and the domain is domain.com. The figure summarizes the steps and where they are carried out.



To configure a clustered environment with a reverse proxy:

1. Confirm that you have the DirectControl agent and the Centrify for Apache package installed as required.
2. If the servers are joined to the domain controller (run `adinfo` to find out), run `adleave` on each Centrify-managed computer to “unjoin.”
3. On machine A, run the following command to join machine A to the domain with aliases for B and C:

   ```
   adjoin -a B-a B.domain.com-a C-a C.domain.comdomain.com
   ```

   Add another -a (--alias) option for each additional Apache server. (See the *Administrator’s Guide for Linux and UNIX* for the description of the adjoin command.)
4. If A has more than one hostname, use the following command to add hostnames:

```
adkeytab -a -P http/other_host_name
```

5. On machine A, run the following commands to replicate the keytabs from machine A onto machines B and C:

   ```
   cd /
   ```

   ```
   tar cvfz cluster.tgz/etc/krb5.keytab/var/centrifydc/kset.*
   scp cluster.tgz B:/
   scp cluster.tgz C:/
   ```

   If you have additional servers, run `scp` to copy `cluster.tgz` to each one.

6. On machines B and C (and each additional server), run the following commands to install the keytabs from machine A and to start `adclient`:

   ```
   cd /
   tar xvfz cluster.tgz
   /usr/share/centrifydc/bin/centrifydc start
   ```

Note If the password for machine A is changed, run Step 5 and Step 6 after every change. This password is changed transparently in a protocol initiated by Active Directory; that is, Active Directory prompts the DirectControl agent for a new account password on an interval defined in the DirectControl agent `adclient.krb5.password.change.interval` configuration parameter (see the *Configuration and Tuning Reference Guide* for the description). The DirectControl agent then automatically generates a new password for the computer account and issues the new password to Active Directory. The default interval is 28 days.

Configure a clustered environment with a load balancer

This section describes how to configure a clustered environment with a load balancer. To provide authentication across all of the servers, you need to create a service account for the load balancer on the domain controller, create a new keytab based on that account, and then merge that keytab on each application server.

Note To create new service accounts, you need permission to the container in which you are creating or deleting the account. See **Understanding object permissions for using adkeytab** in the **Using adkeytab** description in the *Administrator's Guide for Linux and UNIX* for the description of the permissions required.

In this demonstration:

- the DirectControl agent and Centrify for Apache software are already installed on servers B and C (do not install either software package on the load balancer)
- the load balancer hostname is `LB`
- the Apache servers behind the load balancer are named B and C
- the domain is `ace.com`.

The following figure summarizes the steps for a two-server configuration. For each additional machine, perform Step 8 once more on machine B, and Step 9 through Step 16 on each additional machine.

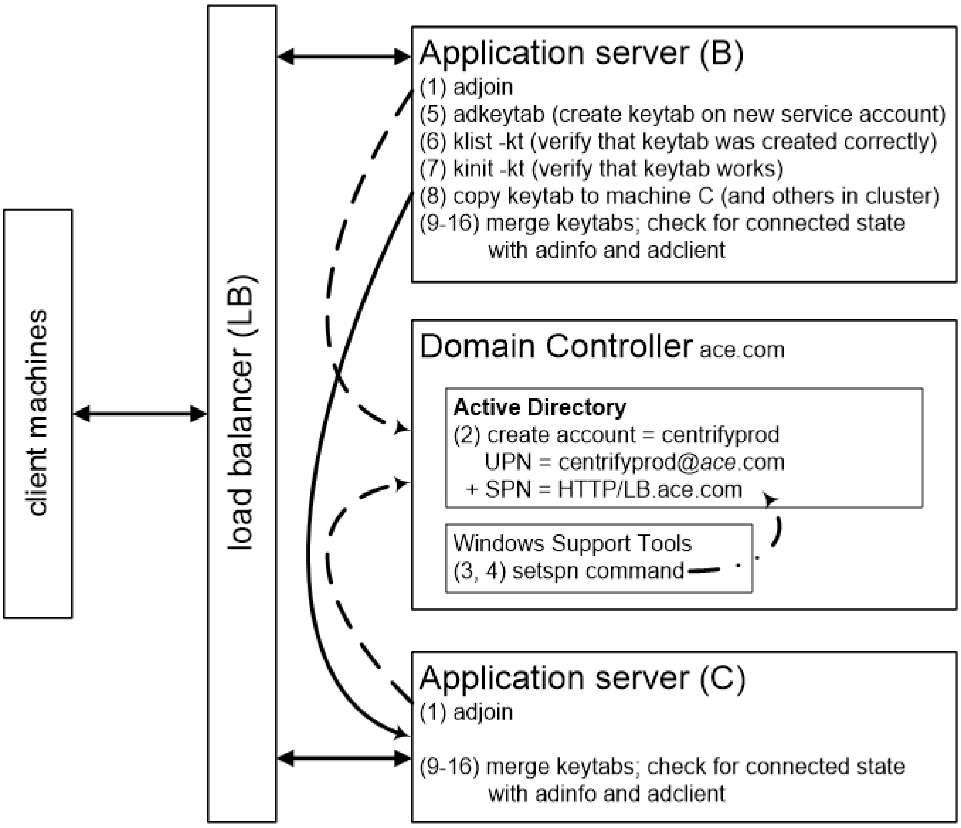
This procedure requires users who have the following permissions:

- Create user account on Active Directory on the domain controller
- Add a new service principal name to the user account on the domain controller
- Change service account password from the UNIX computer.

To configure a clustered environment with a load balancer:

1. Confirm that you have the DirectControl agent (`adclient`) and the Centrify for Web Applications package installed as required.

   Unless they are already joined to the domain controller, run `adjoin` on servers B and C (and all other application servers) to join them to the domain controller.


Application server (B)
(1) adjoin
(5) adkeytab (create keytab on new service account)
(6) klist -kt (verify that keytab was created correctly)
(7) kinit -kt (verify that keytab works)
(8) copy keytab to machine C (and others in cluster)
(9-16) merge keytabs; check for connected state with adinfo and adclient
client machines
load balancer (LB)
Domain Controller ace.com
Active Directory
(2) create account = centrifyprod
UPN = centrifyprod@ace.com
+ SPN = HTTP/LB.ace.com
Windows Support Tools
(3, 4) setspn command
Application server (C)
(1) adjoin
(9-16) merge keytabs; check for connected state with adinfo and adclient

2. Create a new Active Directory account called `centrifyprod`. Verify that the user principal name (UPN) is `centrifyprod@ace.com`.

   Note To have `setspn` available to run in Step 3 and Step 4, you need to install Windows Support Tools.

3. From a Windows system with Windows Support Tools installed, run the `setspn` command to add a new service principal name (SPN) to the user account:

   ```
   setspn -a HTTP/LB.ace.com centrifyprod
   ```

4. Confirm that the SPN was created correctly:

   ```
   setspn -l centrifyprod
   ```

   You should see the SPN `HTTP/LB.ace.com` listed.

   Note Perform Step 5 through Step 8 (below) on machine B *only*.

5. Use the following `adkeytab` command with the `--adopt` option to create the keytab for the new `centrifyprod` account and have the authentication service take over the management of the keytab:

   ```
   adkeytab --adopt --principal HTTP/LB.ace.com \
   --encryption-type arcfour-hmac-md5 \
   --encryption-type des-cbc-md5 \
   ```

```
--encryption-type des-cbc-crc \
--keytab /etc/krb5/centrifyprod.keytab centrifyprod
```

This example uses sample encryption types to illustrate the command. You must make a separate `--encryption-type` entry for each encryption type you use. Replace the options above with the encryption types in your configuration.

Note See the additional information about running `adkeytab` in Notes on running adkeytab.

6. Verify that the keytab was created correctly:

```
/usr/share/centrifydc/kerberos/bin/klist \
  -kt /etc/krb5/centrifyprod.keytab
```

You should see the SPN `HTTP/LB.domain.com`.

7. Verify that the keytab works:

```
/usr/share/centrifydc/kerberos/bin/kinit \
  -kt /etc/krb5/centrifyprod.keytab centrifyprod
```

You should see no output if everything worked correctly.

8. You must have the same Kerberos keytab on each computer. Copy the keytab `/etc/krb5/centrifyprod.keytab` to server C.

Perform Step 9 through Step 16 on both servers B and C.

9. Disable the DirectControl agent to prepare for merging keytabs:

```
svcadm disable centrifydc
```

10. Back up the existing keytab:

```
cp /etc/krb5/krb5.keytab \
  /etc/krb5/krb5.keytab.todaysdate
```

11. Merge the keytabs:

```
/usr/bin/ktutil
rkt /etc/krb5/krb5.keytab
rkt /etc/krb5/centrifyprod.keytab
wkt /etc/krb5/krb5.keytab.new
q
```

12. Verify that the new keytab was created correctly:

```
/usr/share/centrifydc/kerberos/bin/klist \
  -kt /etc/krb5/krb5.keytab.new
```

13. Copy the new keytab to the default location with the appropriate name:

```
cp /etc/krb5/krb5.keytab.new /etc/krb5/krb5.keytab
```

14. Verify that the new keytab works:

```
/usr/share/centrifydc/kerberos/bin/kinit -kt centrifyprod
```

You should see no output if everything worked correctly.

15. Enable the authentication service:

    ```
    svcadm enable centrifydc
    ```

16. Run `adinfo` and check that `adclient` goes into a connected state. If `adclient` reports that it is `disconnected`, something has gone wrong in the setup.

Note If the password for the `centrifyprod` Active Directory account is changed, run Step 5 through Step 16 after every change.This password is changed transparently in a protocol initiated by Active Directory; that is, Active Directory prompts for a new account password on an interval defined in the DirectControl agent `adclient.krb5.password.change.interval` configuration parameter (see the *Configuration and Tuning Reference Guide* for the description). The DirectControl agent then automatically generates a new password for the computer account and issues the new password to Active Directory. The default interval is 28 days.

Notes on running adkeytab

To run this `adkeytab` command the user must have write permission to change the password for the service account and read/write permission to the `userAccountControl` attribute on the Active Directory domain controller. (See **Understanding object permissions for using adkeytab** in the **Using adkeytab** description in the *Administrator's Guide for Linux and UNIX* for the description of the permissions required.) Often, this is NOT the case for the UNIX administrator running `adkeytab`.

Use the following `adkeytab` option to work around this problem. This does require, however, the UNIX admin to know and then expose the password in the command line. (The alternative would be to give the Active Directory admin root privileges on the Linux or UNIX computer or the UNIX admin password reset privileges on the domain controller.)

- The Active Directory administrator creates the new AD account and adds the SPN to the account as above but then provides the password to the UNIX admin.
- The UNIX admin uses the following `adkeytab` command instead of the command in Step 5. In this example the new user created by the AD

admin is again `centrifyprod@ace.com` and the password is `ABC123xyz`:

```
adkeytab --adopt --user centrifyprod@ace.com \
--local --newpassword ABC123xyz \
--encryption-type arcfour-hmac-md5 \
--encryption-type des-cbc-md5 \
--encryption-type des-cbc-crc \
--keytab /etc/krb5/centrifyprod.keytab centrifyprod@ace.com
```

Note The `--user` option specifies the new account created by the AD admin; `--local` updates the keytab file on the computer (in this case, B) without changing the password in AD and `--newpassword` specifies the new password (required by the `--local` option). (This example uses the same sample encryption types as above.) See the `adkeytab` description in the *Administrator's Guide for Linux and UNIX* for the full explanation of each option.