



## Configuring Apache Web Server For Single Sign-On with Likewise

### IN THIS DOCUMENT

- Setting up an Apache HTTP Server for single sign-on with Likewise, Kerberos, and Active Directory.
- Understanding Integrated Windows Authentication.
- Likewise's Apache authentication architecture.
- Installing and configuring the `mod_auth_kerb` authentication modules.
- Generating a keytab file.
- Testing authentication.
- Troubleshooting authentication.
- Dealing with common issues.

### REQUIREMENTS

- Apache HTTP Server version 2.0 or 2.2.
- The Linux or Unix computer running Apache must be using a platform that Likewise supports.
- Root access to the Linux or Unix computer running Apache.
- Privileges sufficient to join the Linux or Unix computer running Apache to Active Directory.

### Abstract

Likewise Enterprise lets you join Linux and Unix computers running the Apache HTTP Server to Microsoft Active Directory, yielding a range of benefits for users, system administrators, and managers.

Users get single sign-on: They log on once to a workstation that is authenticated through Active Directory and automatically receive Kerberos-based single sign-on for other computers and applications, including the Apache web server. System administrators rest easy with the knowledge that users accessing your intranet through HTTP are securely authenticated with Kerberos 5 and authorized for access to the resources on your Apache web server. Managers see their operational costs drop as their Linux and Unix computers running Apache are centrally managed within Active Directory. Security managers find help in their quest for regulatory compliance.

[This document describes how to configure Likewise and the Apache HTTP Server to provide single sign-on authentication through Kerberos.](#)

### About Likewise Enterprise

By joining Linux, Unix, and Mac computers to Active Directory – a secure, scalable, stable, and proven identity management system – Likewise gives you the power to manage all your users' identities in one place, use the highly secure Kerberos 5 protocol to authenticate users in the same way on all your systems, apply granular access controls to sensitive resources, and centrally administer Linux, Unix, Mac, and Windows computers with group policies. Likewise includes reporting and auditing capabilities that can help improve regulatory compliance. The result: lower operating costs, better security, enhanced compliance.

The information contained in this document represents the current view of Likewise Software on the issues discussed as of the date of publication. Because Likewise Software must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Likewise, and Likewise Software cannot guarantee the accuracy of any information presented after the date of publication.

These documents are for informational purposes only. LIKewise SOFTWARE MAKES NO WARRANTIES, EXPRESS OR IMPLIED.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form, by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Likewise Software.

Likewise may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Likewise, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2008 Likewise Software. All rights reserved.

Likewise and the Likewise logo are either registered trademarks or trademarks of Likewise Software in the United States and/or other countries. All other trademarks are property of their respective owners.

Likewise Software  
15395 SE 30th Place, Suite #140  
Bellevue, WA 98007  
USA



## Table of Contents

<b>INTRODUCTION</b> .....	<b>4</b>
<b>UNDERSTANDING INTEGRATED WINDOWS AUTHENTICATION</b>	
<b>    AUTHENTICATION</b> .....	<b>4</b>
Why use Integrated Windows Authentication? .....	4
Kerberos, NTLMSSP versus Basic Authentication .....	5
Authentication versus Authorization .....	5
<b>    LIKEWISE APACHE AUTHENTICATION ARCHITECTURE</b> .....	<b>5</b>
Authenticating with Kerberos.....	6
Authenticating with Basic Authentication.....	6
<b>CONFIGURING LIKEWISE AND APACHE FOR SSO</b> .....	<b>6</b>
Overview of Setup Process.....	6
Requirements .....	7
Check Whether Your Apache Server Supports Dynamic Modules .....	7
<b>Install mod_auth_kerb_centeris and mod_auth_pam Authentication Modules</b> .....	<b>8</b>
32-Bit Systems .....	8
64-Bit Systems .....	9
Install the Packages .....	9
<b>Add Directives to Apache Configuration File</b> .....	<b>9</b>
<b>Setup Apache Server or Virtual Server To Use SSL (Optional)</b> .....	<b>10</b>
<b>Generate Kerberos keytab File</b> .....	<b>10</b>
Find the Group Name of the Apache Web Server .....	10
Find the Server Name of the Web Site that Requires Authentication .....	11
Generate a Kerberos keytab file .....	12
<b>Set mod_auth_kerb_centeris.so and mod_auth_sys_group.so</b> .....	<b>12</b>
<b>TESTING AUTHENTICATION</b> .....	<b>14</b>
Test Single-Sign-On authentication for a domain user .....	14
Test the Basic Authentication Fallback.....	14
<b>TROUBLESHOOTING</b> .....	<b>15</b>
Apache Log File .....	15
The Microsoft KERBTRAY utility.....	15
KLIST Linux or Unix Utility.....	16
Common Issues .....	17
Contact Technical Support .....	18

## Introduction

This document describes how to configure Likewise and the Apache HTTP Server to provide single sign-on authentication through Active Directory by using Kerberos 5.

The target audience is network directory administrators who manage access to workstations, servers, and other network resources within Active Directory. The guide assumes that you know how to administer Active Directory as well as computers running Linux.

**Note:** This guide assumes that you have installed Likewise on the Linux computer running your Apache HTTP Server and that you have joined the server to Active Directory. For instructions on how to install Likewise on a Linux computer and join it to Active Directory, see the Likewise Enterprise Installation Guide at [http://www.likewisesoftware.com/resources/user\\_documentation/](http://www.likewisesoftware.com/resources/user_documentation/).

## Understanding Integrated Windows Authentication

Integrated Windows Authentication was introduced with the Microsoft Windows 2000 operating system. It is based on the SPNEGO, Kerberos, and NTLMSSP protocols. The SPNEGO protocol is used between the web browser and the web server to negotiate the type of authentication that will be performed, usually either Kerberos or NTLMSSP. Kerberos is the preferred authentication mechanism. Both Kerberos and NTLMSSP are secure protocols that allow computers to authenticate a user over a non-secure channel. For web sites, this means that the Secure Socket Layer (SSL) protocol does not need to be enabled during the authentication phase.

### Why use Integrated Windows Authentication?

Integrated Windows Authentication improves the overall security of a network because the user must log on by using his or her username and password only once. All subsequent accesses by that user to resources, such as web sites, file systems, and network printers are automatically authenticated with cached security tokens. Using Integrated Windows Authentication has the benefit of a centralized user account database where information about all users is kept in Active Directory. This is more secure than duplicating user names and passwords in configuration files



across various server computers, not to mention the management overhead of doing so.

#### **Kerberos, NTLMSSP versus Basic Authentication**

Integrated Windows Authentication uses the SPNEGO, Kerberos and NTLM authentication protocols. Not all browsers are capable of understanding these protocols. Another authentication protocol, Basic Authentication, is understood by all web browsers; it works by simply transferring username and password across the network from the web browser to the web server. The drawback of using Basic Authentication is that without SSL encryption, anyone can intercept the network communication and easily find out a user's login name and password. Thus, Basic Authentication should be used only for sites that are protected with SSL encryption.

#### **Authentication versus Authorization**

The term *authentication* refers to the process of proving a user's identity. *Authorization*, on the other hand, takes place after authentication and is used to limit the users or groups to perform only the actions that they are allowed, or authorized, to perform. Integrated Windows Authentication provides only the authentication mechanism while Windows authorization is usually accomplished using Access Control Lists (ACL).

## Likewise Apache Authentication Architecture

The Likewise Apache Authentication architecture extends Integrated Windows Authentication to the Apache web server running on a Linux or Unix system. The authentication is implemented in a dynamically loaded Apache module: `mod_auth_kerb_centeris`. This module is based on a BSD licensed Apache module called `mod_auth_kerb`, but includes modifications so that it works with the Likewise.

An additional module — `mod_auth_sys_group` — is used to provide authorization limiting access to the web site to the domain users or groups that you specify.

The `mod_auth_kerb_centeris` module implements the SPNEGO, Kerberos, and Basic Authentication protocols. In doing so, it provides the majority of the Integrated Windows Authentication functionality, with the exception of the NTLMSSP protocol. The module uses the SPNEGO protocol to negotiate whether Kerberos or Basic Authentication is used.



### Authenticating with Kerberos

If the client browser is able to obtain a Kerberos ticket from the Active Directory domain controller for the requested URL, Kerberos authentication is used and the user does not need to provide his or her username and password to access the web site. The Kerberos ticket is sent to the web server, which decrypts it and validates it by using the Kerberos system libraries.

The Kerberos ticket contains the user's *principal name*, which being created by a Windows domain controller is in the form `<user>@<full domain name>`; **example:** `user@mydoomain.com`.

The principal name is converted by `mod_auth_kerb_centeris` to a format — `<short domain name>\<user>` — needed by the `mod_auth_sys_group` authorization module, **example:** `MYDOMAIN\user`. If the authorization is successful, the server sends the requested web page to the client, but with additional authentication information that allows the web browser to validate the authenticity of the web server for mutual authentication.

### Authenticating with Basic Authentication

If the client browser is not able to obtain a Kerberos ticket from the Active Directory Controller and `mod_auth_kerb_centeris` is configured with the `KrbMethodK5Passwd` parameter set to "on" then Basic Authentication is used. The browser prompts the user for username and password and sends this information in clear-text to the web server. The web server tries to obtain a Kerberos ticket on user's behalf directly from the Domain Controller. If successful, it processes the Kerberos ticket and performs authorization. If the Authorization is successful, the server sends the requested web page to the client, but unlike with full Kerberos authentication, it does not include the additional authentication information that allows the client to authenticate the web server's identity.

## Configuring Likewise and Apache for SSO

### Overview of Setup Process

1. Confirm that your components meet the requirements.
2. Install the `mod_auth_kerb_centeris` and `mod_auth_pam` Apache authentication modules.



3. Configure the main Apache server or Virtual Host to use SSL (optional).
4. Generate a Kerberos keytab file for the Apache server.
5. Configure the `mod_auth_kerb_centeris.so` and `mod_auth_sys_group.so` modules.

#### Requirements

- At minimum, setting up Apache SSO authentication requires two systems, a Linux or Unix computer running the Apache HTTP Server and an Active Directory domain controller.
- You must have root access to a Linux or Unix system that is supported by Likewise; for a list of supported platforms, see [http://www.likewise.com/products/likewise\\_enterprise/supported\\_platforms.php](http://www.likewise.com/products/likewise_enterprise/supported_platforms.php).
- You must have installed the Likewise agent on the Linux or Unix computer and joined the computer to Active Directory; for instructions on how to install the Likewise agent and join the server to a domain, see Likewise Enterprise Installation Guide at [http://www.likewise.com/resources/user\\_documentation/](http://www.likewise.com/resources/user_documentation/)
- The supported versions of Apache HTTP Server are 2.0 and 2.2. It is recommended that you use the Apache version that came with your Linux or Unix system. If you need to install the Apache server manually, you can get it and installation instructions at <http://httpd.apache.org>.
- The Apache HTTP Server that you are using must support dynamic loading of the Apache modules in order to load the `mod_auth_kerb_centeris` module.

#### Check Whether Your Apache Server Supports Dynamic Modules

1. Execute the following command on the Linux or Unix system after the Apache HTTP Server has been installed:

```
# httpd -l
```

Depending on the system, the command may be `/usr/sbin/httpd -l`, `/usr/sbin/httpd2 -l` or in another location.



2. Verify that `mod_so.c` is displayed in the list of the compiled in modules.

**Important:** For Apache installations that are compiled from the source code, ensure that `--enable-module=so` is specified when `./configure` is executed:

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

### Install `mod_auth_kerb_centeris` and `mod_auth_pam` Authentication Modules

The `mod_auth_kerb_centeris` and `mod_auth_pam` Authentication modules are installed from RPM packages. The packages are specific to each Linux or Unix operating system, but may also vary if the Apache web server was upgraded from the version shipped with the Linux or Unix system.

The following tables list the RPM package names for each supported system.

#### 32-Bit Systems

Operating System	RPM packages
Red Hat Enterprise Linux ES / AS v4 CentOS 4.2, 4.1 CentOS 4.4, 4.3 Fedora Core 4 (with full updates)	<code>mod_auth_kerb_centeris-5.0-6.redhat.i386.rpm</code> <code>mod_auth_pam_rhel-1.0-1.i386.rpm</code>
Red Hat Enterprise Linux ES / AS v3	<code>mod_auth_kerb_centeris-5.0-6.rhel3.i386.rpm</code> <code>mod_auth_pam-1.1.1-1.rhel3.i386.rpm</code>
Fedora Core 5	<code>mod_auth_kerb_centeris-5.0-6.fc5.i386.rpm</code> <code>mod_auth_pam-1.1.1-4.fc5.i386.rpm</code>
SUSE Linux Enterprise Server 9 (SP3) SUSE Linux Professional 9.2	<code>mod_auth_kerb_centeris-5.0-6.suse92.i586.rpm</code> <code>mod_auth_pam_suse-1.0-1.i586.rpm</code>
SUSE Linux Enterprise Server 10 SUSE Linux 10.1	<code>mod_auth_kerb_centeris-5.0-6.suse.apache2.2.i586.rpm</code> <code>mod_auth_pam_suse-1.0-1.apache2.2.i586.rpm</code>
SUSE Linux Professional 9.3 SUSE Linux 10.0 SUSE Linux 10.0 OSS	<code>mod_auth_kerb_centeris-5.0-6.suse.i586.rpm</code> <code>mod_auth_pam_suse-1.0-1.i586.rpm</code>

## 64-Bit Systems

Operating System	RPM packages
Red Hat Enterprise Linux ES / AS v4 CentOS 4.4, 4.3 Fedora Core 4 (with full updates)	mod_auth_kerb_centeris-5.0-6.redhat.x86_64.rpm mod_auth_pam-1.1.1-4.x86_64.rpm
Red Hat Enterprise Linux ES / AS v3	mod_auth_kerb_centeris-5.0-6.rhel3.x86_64.rpm mod_auth_pam-1.1.1-1.rhel3.x86_64.rpm
Fedora Core 5	mod_auth_kerb_centeris-5.0-6.fc5.x86_64.rpm mod_auth_pam-1.1.1-4.fc5.x86_64.rpm
SUSE Linux Enterprise Server 9 (SP3)	mod_auth_kerb_centeris-5.0-6.suse_heimdal.x86_64.rpm mod_auth_pam-1.1.1-1.x86_64.rpm
SUSE Linux Enterprise Server 10 SUSE Linux 10.1	mod_auth_kerb_centeris-5.0-6.suse.apache2.2.x86_64.rpm mod_auth_pam-1.1.1-1.apache2.2.x86_64.rpm

Contact Likewise support at <http://www.likewise.com/support/> to obtain the modules for your platform.

### Install the Packages

To install the `mod_auth_kerb_centeris` and `mod_auth_pam` packages, execute the following commands substituting the `<version>` for the correct version for your system:

```
# rpm -i mod_auth_kerb_centeris-5.0-6.<version>
# rpm -i mod_auth_pam<version>
```

**Note:** The `mod_auth_pam` RPM package contains two modules. Only one of the two modules, `mod_auth_sys_group.so`, is necessary to complete the setup documented in this guide.

The Apache web server configuration must now be modified to load the `mod_auth_kerb_centeris.so` and `mod_auth_sys_group.so` modules. The location of the Apache configuration file will depend on your Linux or Unix distribution.

### Add Directives to Apache Configuration File

Add the following directives to the Apache configuration file to configure Apache to load the `mod_auth_kerb_centeris.so` and `mod_auth_sys_group.so` modules.



Platforms	Directives
Red Hat	LoadModule auth_kerb_module modules/mod_auth_kerb_centeris.so
Linux	LoadModule auth_sys_group_module modules/mod_auth_sys_group.so
Novell SuSE	LoadModule auth_kerb_module /usr/lib/apache2-prefork/mod_auth_kerb_centeris.so
Linux	LoadModule auth_sys_group_module /usr/lib/apache2-prefork/mod_auth_sys_group.so

## Setup Apache Server or Virtual Server To Use SSL (Optional)

It is recommended that all web sites that require authentication communicate over the Secure Socket Layer (SSL) protocol. SSL encrypts all data that passes between the client browser and the web server. Furthermore, SSL allows performing Basic Authentication as a fallback mechanism in a secure fashion. Basic Authentication is the only fallback mechanism provided by `mod_auth_kerb_centeris` in the case when the primary, Kerberos authentication does not succeed. This is especially important if the protected website also needs to be accessible from outside the corporate network.

Configuring SSL for the main Apache server or virtual server is beyond the scope of this guide. Various installations of the Apache web server provide sample SSL configuration files that can be included from the main `httpd.conf` file. In most cases you will need to generate a self-signed SSL certificate or for production environments purchase a commercial certificate. On most Linux or Unix systems a self signed certificate can be generated using the `openssl` command.

## Generate Kerberos keytab File

The Kerberos keytab file is necessary to authenticate incoming requests. It contains encrypted, local copy of the host's key and if compromised may allow unrestricted access to the host computer. It is therefore crucial to protect it with the appropriate file access permissions. This file must be readable by the user group under which the Apache web server is running, usually `apache` on Red Hat systems or `www` on SuSE systems.

### Find the Group Name of the Apache Web Server

1. Locate the main Apache `httpd.conf` configuration file.
2. Search the `httpd.conf` file for the `Group` directive (case sensitive).

Example: Group apache

Next, you will need to find out the server name of the web site that will require authentication. The authentication will actually protect a specific directory, thus restricting access to a physical location on the file system. The directory to protect will either be in the context of the main Apache server or in the context of a Virtual Host.

#### Find the Server Name of the Web Site that Requires Authentication

1. Locate the Apache configuration file that contains the `<Directory>` directive of the physical directory to be protected by authentication. Depending on your setup this may be in the main `httpd.conf` configuration file or a file included from the main file.
2. Locate the `ServerName` directive within the `<VirtualHost>` container containing the `<Directory>` directive or locate the global `ServerName` directive if not using `<VirtualHost>`.

Example:

```
<VirtualHost *:80>
ServerName myserver:80
DocumentRoot /srv/www/htdocs
<Directory /srv/www/htdocs>
AllowOverride None
Options -Indexes
Order allow,deny
Allow from all
</Directory>
```

The server name in the example above is `myserver`.

**Note:** If the `ServerName` is not specified in the configuration file, try doing a reverse DNS name lookup on the IP address of the host or just use the hostname of the Linux or Unix system.

You will also need to know the full domain name of the domain to which the Linux or Unix system is joined. Finally, you will need to figure out where to save the generated keytab file. It is possible to keep one keytab

file for all Virtual Host sites on your server or to create a separate keytab file for each Virtual Host.

### Generate a Kerberos keytab file

The steps below use a sample Apache user account name: apache, a sample server name: myserver, a sample full domain name: MYDOMAIN.COM and a sample Kerberos keytab file name: /etc/apache2/protected/krb5\_myserver.keytab. Substitute your own names for these values.

1. Set the KRB5\_KTNAME environment variable to point to the Kerberos keytab file to be generated.  
**# export**  
**KRB5\_KTNAME=FILE:/etc/apache2/protected/krb5\_myserver.keytab**
2. Generate keytab entry for the default HTTP service principal.  
**# /usr/centeris/bin/lwinet ads keytab add HTTP -P**
3. Generate keytab entry for myserver service principal  
**# /usr/centeris/bin/lwinet ads keytab add**  
**HTTP/myserver@MYDOMAIN.COM -P**
4. Generate keytab entry for myserver.mydomain.com HTTP service principal  
**# /usr/centeris/bin/lwinet ads keytab add**  
**HTTP/myserver.mydomain.com@MYDOMAIN.COM -P**
5. Change the group ownership of the keytab file  
**# chown root.www**  
**/etc/apache2/protected/krb5\_myserver.keytab**
6. Set appropriate file permissions of the keytab file  
**# chmod 640 /etc/apache2/protected/krb5\_myserver.keytab**

### Set mod\_auth\_kerb\_centeris.so and mod\_auth\_sys\_group.so

The only configuration that remains is to modify the Apache configuration file by adding appropriate directives in each <Directory> container that is to be protected.

To enable single sign-on authentication, add the following directives inside the protected <Directory> container:

**Note:** Replace `/etc/apache2/protected/krb5_myserver.keytab` with your own keytab file name and replace `MYDOMAIN` with your short domain name.

```
AuthType Kerberos
Krb5Keytab
/etc/apache2/protected/krb5_myserver.keytab
KrbMethodK5Passwd on
AuthName "Secure Server"
require group "MYDOMAIN\domain users"
require user "MYDOMAIN\Administrator"
```

The following table explains the meaning of the commonly used directives for `mod_auth_kerb_centeris.so` and `mod_auth_sys_group.so` modules.

Directive Name	Values	Explanation
AuthType	"Kerberos"	This value should always be set to "Kerberos".
Krb5Keytab	file path	Points to the location of the Kerberos V5 keytab file readable by Apache.
KrbMethodK5Passwd	"on" or "off"	Enables or disables Basic Authentication fallback mechanism. Should only be set to "on" if SSL is enabled.
AuthName	quoted string	The name of the authorization realm for a directory. This realm is given to the client so that the user knows which username and password to send.
require user	space delimited user list	Only the named users can access the resource.
require group	space delimited group list	Only users in the named groups can access the resource.

To complete the configuration, you must restart the Apache web server:



```
Red Hat Linux    # /etc/init.d/httpd restart
Novell SuSE     # /etc/init.d/apache2 restart
Linux
```

## Testing Authentication

The first test is to determine if Integrated Windows Authentication is working for all domain users. The require group “MYDOMAIN\domain users” directive will allow any user that is part of the “domain users” group to authenticate.

**Note:** Use Internet Explorer because it supports Integrated Windows Authentication. Verify that the “Enable Integrated Windows Authentication” checkbox is selected in the Internet Explorer Internet Options dialog on the Advanced tab.

### Test Single-Sign-On authentication for a domain user

1. Logon to a Windows computer that is joined to the same domain you joined your Linux or Unix system to. Logon as a domain user.
2. Access the protected web site from Internet Explorer.  
NOTE: Do not access the site using the IP address of the server because doing so will not result in the Internet Explorer getting a Kerberos ticket for the server. Use the server name instead.  
Example: https://myserver

The authentication should succeed without a need to provide a user name and password.

### Test the Basic Authentication Fallback

The second test is to determine if the Basic Authentication fallback mechanism is working. This test will only work if the KrbMethodK5Passwd directive was set to “on” and for production environments should only be done over SSL.

1. Logon to a Windows computer using a local user account. Alternatively, you can logon to a computer that is joined to a different domain than the Linux or Unix Web server. Another possibility is to use another browser that does not support



Integrated Windows Authentication.

2. Access the protected web site from the browser.  
Example: `https://myserver`
3. You should be prompted for user name and password.

The authentication should succeed after providing a domain user name and password. Be sure to prefix the user name with the short domain name. For example: `MYDOMAIN\Administrator`.

## Troubleshooting

In the case of authentication failure, there are some diagnostic tools that may help diagnose a problem.

### Apache Log File

The Apache log file may contain helpful information as to the nature of the problem. Locate your Apache log file by examining the `ErrorLog` directives in the Apache configuration file. You can increase the logging detail by setting the `LogLevel` directive to “debug”.

### The Microsoft KERBTRAY utility

The Microsoft KERBTRAY utility, part of Microsoft Windows 2000 Resource Kit, will help to determine if the Internet Explorer browser obtained a Kerberos ticket for your web server. Integrated Windows Authentication requires that the client obtains a Kerberos ticket from the Active Directory. To check if the web browser obtained a Kerberos ticket for your web server:

1. Logon to a Windows computer that is joined to the domain you joined your Linux or Unix system to. Logon as a domain user.
2. Install the Microsoft KERBTRAY utility. The setup should be available at the following URL  
`http://download.microsoft.com/download/win2000platform/kerbtray/1.0.0.0.1/NT5/EN-US/kerbtray_setup.exe.`
3. Launch `KERBTRAY.EXE`. This will install an icon in your taskbar tray.

4. Access the protected web site from Internet Explorer using the server name in the URL rather than the IP address.  
Example: `https://myserver`
5. Double click on the KERBTRAY icon in the taskbar tray. This will show a dialog displaying all Kerberos tickets.
6. Look for the name of your domain in the list of the tickets. Under your domain look for service principal that will look like `HTTP/myserver.mydomain.com`.

### KLIST Linux or Unix Utility

The `klist` Linux or Unix utility, part of the `krb5-client` package, may be used to check the Kerberos keytab file on the Linux or Unix system. The output of this command will display all service principal tickets that are contained in the keytab file. This command may be unavailable on some systems. To use `klist` to examine the contents of a Kerberos keytab file:

**NOTE:** Replace `myserver` with your own server name and `mydomain.com` with your domain name.

1. Locate the Kerberos keytab file for the protected web site. You may need to find the `Krb5Keytab` directive in the Apache configuration file.
2. Execute the `klist -k <keytab file name>` command.  
**# `klist -k krb5_myserver.keytab`**
3. Verify that the correct service principal names are displayed. The names to look for are `HTTP/myserver@MYDOMAIN.COM` and `HTTP/myserver.mydomain.com@MYDOMAIN.COM`. It is normal to see multiple entries for the same name.

Example output:

```
# klist -k krb5_myserver.keytab
Keytab name: FILE:krb5_myserver.keytab
KVNO Principal
-----
 6 HTTP/myserver@MYDOMAIN.COM
 6 HTTP/myserver@MYDOMAIN.COM
 6 HTTP/myserver@MYDOMAIN.COM
 6 HTTP/myserver.mydomain.com@MYDOMAIN.COM
```

6 HTTP/myserver.mydomain.com@MYDOMAIN.COM

6 HTTP/myserver.mydomain.com@MYDOMAIN.COM

If the service principal names are not correct, go back to the Generate Kerberos keytab file step above to generate a new Kerberos keytab file.

### Common Issues

As Authentication problems may be difficult to diagnose, start with double checking all the configuration parameters including the validity of the generated Kerberos keytab file. If all the configuration parameters appear to be correct then examine the list of common problems below.

Problem	Explanation
System clock out of sync	For Kerberos authentication to work the system clocks on all involved systems must not be more than 5 minutes apart. Make sure that the time on the Active Directory server, Linux or Unix web server and the client are synchronized.
User accessing the web site is not on the "require" list	If Kerberos ticket was obtained on the client or the user correctly entered his credentials during Basic Authentication prompt it may be the case the authentication worked but the authorization failed. In this case the error_log Apache log file will contain a line similar to the one below.  access to / failed, reason: user MYDOMAIN\user not allowed access  Add the user to the "require user" directive or add the user's group to "require group" directive.
User accessing the web site is logged to wrong domain	If the client user is logged into a domain different than the domain of the web server then one of two things will happen. If the KrbMethodK5Passwd directive is set to "on" then the user will be prompted for credentials. If KrbMethodK5Passwd is set to "off" the authentication will fail and the "Authorization Required" page will be displayed.
Internet Explorer web browser does not consider the URL as part of the Local Intranet zone.	This problem is common if the web site is accessed using a URL that includes the full domain name such as https://myserver.mydomain.com. Internet Explorer will only try to obtain Kerberos tickets for websites that are in the Local Intranet zone. Try accessing the web site using just the server name, for example https://myserver. Alternatively, you can add the URL to a list of Local Intranet sites using the Sites/Advanced buttons in the Internet Options dialog on the Security tab.

<p>Service Principal Name of the web site is mapped to more than one object in the Active Directory</p>	<p>This is a rather rare problem, but very difficult to diagnose because of lack of good error that is returned to the web server. This can happen if the ktpass Windows utility was used on the Domain Controller to generate a Kerberos keytab file.</p> <p>To diagnose this problem log onto the Active Directory domain controller and open the Event Viewer. Look for event of type=Error, source=KDC, and event ID=11. The text of the event will be similar to the below message.</p> <p>There are multiple accounts with name HTTP/myserver.mydomain.com of type DS_SERVICE_PRINCIPAL_NAME.</p> <p>Fixing this problem will require locating the computer or user objects used to map the service principal name in the Active Directory. The Active Directory Users and Computers MMC snap-in allows running custom LDAP queries. Use a LDAP query similar to this one “(servicePrincipalName=HTTP/myserver.mydomain.com)” to locate the Active Directory objects. Once object have been located then the spurious User object may be deleted. If the object cannot be deleted then use the ADSI Edit MMC snap-in to manually remove the “HTTP/myserver.mydomain.com” string from the servicePrincipalName object property.</p>
---	--

### Contact Technical Support

For either post-sales technical support or for free technical support during an evaluation period, please visit the Likewise support web page at <http://www.likewise.com/support/>. You can use the support page to register for support, submit incidents, and receive direct technical assistance.

Technical support may ask for your Likewise version, Linux version, and Microsoft Windows version. To find the Likewise product version, in the Likewise Console, on the menu bar, click **Help**, and then click **About**.

### ABOUT LIKEWISE

Likewise Software is an open source company that provides audit and authentication solutions designed to improve security, reduce operational costs and help demonstrate regulatory compliance in mixed network environments. Likewise Open allows large organizations to securely authenticate Linux, UNIX and Mac systems with a unified directory such as Microsoft Active Directory. Additionally, Likewise Enterprise includes world-class group policy, audit and reporting modules.

Likewise Software is a Bellevue, WA-based software company funded by leading venture capital firms Ignition Partners, Intel Capital, and Trinity Ventures. Likewise has experienced management and engineering teams in place and is led by senior executives from leading technology companies such as Microsoft, F5 Networks, EMC and Mercury.