



OpenSSH & Kerberos

*Prepared By
Visolve SSH Team*

| [Abstract](#) | [Authentication and Authorization](#) | [Overview of Kerberos](#) | [Overview of SSH](#) |
| [Common Definitions](#) | [Configuring Kerberos Environment](#) | [Installing Kerberos](#) | [Configuring the KDC](#) |
| [Configuration on Application Server Host](#) |
[Configuration on Application Client Host](#)	[OpenSSH - GSSAPI Authentication](#)	
[OpenSSH - Kerberos Authentication](#)	[RFCs for Kerberos](#)	[References](#)
[Support](#)	[About Visolve.com](#)	

Abstract

Internet connects tens of millions of computers across the globe. The Internet Protocol suite, including the TCP/IP, provides reliable and scalable communications over the real-world networks. However, as Information espionage has become a major problem in Internet now-a-days, this vast network is prone/vulnerable to many malicious attacks such as eavesdropping, connection hijacking, IP spoofing, etc. Using Secure Shell with Kerberos authentication is an ideal choice for preventing these types of network risks and threats. This paper discuss about setting up of Kerberos environment and how to configure the Kerberos and Secure Shell to ensure secure and reliable communication over the network. There are two versions of Kerberos

available, Kerberos4 and Kerberos5. This guide discuss only on Kerberos5. Those who want to know the introduction to Kerberos and OpenSSH, may move on to the next section. If you want to look at the configuration of Kerberos and OpenSSH, you can skip this section and directly go to [Configuring Kerberos environment](#) for setting up the Kerberos environment on Linux with OpenSSH. For setting up a Kerberos environment on HP-UX with HP-UX Secure Shell, see, [Setting up a Kerberos environment for HP-UX Secure Shell](#)

Authentication and Authorization

In any network communication, the two stages that the user has to pass through, for getting connected to a remote system and getting access to the resources on the remote system are "Authentication" and "Authorization".

- Authentication is "the process of verifying an identity claimed by or for a system entity". This is to provide assurance that users (or systems) are who they say they are. Thus, authentication process compares, what user has, to a known constant that is trusted. Most of the time this is done by interacting with the user like getting the username and password. Other forms with the by proxy such as smartcard where the authentication tokens will be stored. These standards are described in [RFC 2828](#) (Internet Security Glossary, May 2000).
- Authorization refers to a user's ability to access resources on a network, based on user account privileges and rights. This is also known as "access control". Only a user who succeeded in authentication will enter to this authorization phase.

Overview of Kerberos

Kerberos, developed and released by MIT, is an AA [Authentication and Authorization] system. Kerberos has two purposes, security and authentication. In most distributed network system, most of the time, password is used to prove user's identity, and password must be transmitted over the network from the client machine to the machine that the user wants to access. So, a mechanism that prevent anyone from intercepting or eavesdropping on the transmitted plain passwords is vital for security. Also another pain in using password for the authentication is, the password has to be supplied each and every time a connection is requested to the remote machine. Kerberos help users to avoid this pain and the central problem solved by Kerberos is how to use passwords for authentication without sending them over the network.

The following infrastructure are required for setting up a Kerberos environment

- **Key Distribution Centre [KDC]:** KDCs are central to the Kerberos system and this system should be heavily secured. This system should run KDC only and other remote logins should be denied. If at all it is desired to allow remote login, SSH is preferred. This system should be kept in a physically secure location.
- **Slave KDC [For backup]:** Kerberos cannot operate without a KDC. So, it is good to have a backup system for the main KDC and this can be act as a slave. This host will synchronize periodically with the master KDC and if the master host fails, this slave will takeover the control.
- Once a KDC host is ready, proper configurations should be done for using this

with OpenSSH. More details are given here - [KDC configuration](#)

Although Kerberos provides some minimal level of authorization facilities, it is limited only for the permissions that the user can have to modify the KDC database. Thus, the individual applications need to take care of the authorization part.

This document is mainly for guiding users in configuring Kerberos to use with OpenSSH. For more information about Kerberos and other terms discussed about Kerberos, click here - "[How Kerberos authentication works?](#)".

Overview of OpenSSH

OpenSSH, primarily developed by the OpenBSD Project, is a free version of the SSH protocol suite of network connectivity tools. OpenSSH encrypts all traffic (including passwords) to effectively eliminate eavesdropping, connection hijacking, IP spoofing and other network level attacks. This distribution provides many authentication methods and one among that is the Kerberos authentication.

OpenSSH and HP-UX Secure Shell supports SSH Protocol versions 1.3, 1.5 and 2.0. As per the current version of OpenSSH [4.3p2],the Kerberos authentication is done via their Kerberos credentials and the authenticated users are allowed to forward their credentials to a remote machine over ssh. Kerberos authentication support is available for SSH protocols 1 and 2. For SSH protocol 2.0, GSSAPI support is also available. In addition to other authentication mechanism support, GSSAPI facilitates authentication with Kerberos also.

Common Definitions

Before getting into the details of configuration, some of the definitions to be known.

Realm: The realm is an administrative division identifying a single instance of Kerberos principal database. Each host is given a realm name. If you are maintaining a KDC for your domain xyz.org, then the realm is the name for the host on which this KDC service is running. Realm names are usually in capital letters [normal convention]. So, the realm name for your example domain is XYZ.ORG. This realm will have a database that holds the user principals.

Principal: A principal is essentially a username for the realm. Each user needs a principal in the realm. This is of the form, principal_name@REALM. For example, ssteam@VISOLVE.COM. In addition to a principal, you must have an account on each machine in the network that comes under this realm, that you will use. If the principal name and the account name are the same, then there are more conveniences.

Ticket: A record that helps a client authenticate itself to a server; it contains the client's identity, a session key, a timestamp, and other information, all sealed using the server's secret key. It only serves to authenticate a client when presented along with a fresh Authenticator.

TGT: Ticket Granting ticket is issued by AS [Authentication Server - a component in KDC], to client for the desired application service, which is encrypted by the client's secret key.

Configuring the Kerberos environment

The following are the requirements for setting up Kerberos on Linux [to use with OpenSSH]
A KDC, Application Server and an Application Client. We have used three Linux machines to demonstrate this setup. The OpenSSH client and Server machines are referred as Application client and Application Server machines respectively.
Kerberos should be installed on all the three machines
All the machines should contain krb5.conf file in the /etc directory and make sure that the krb5.conf file is common for all the 3 machines.
DNS should be configured for all the server and client host entries.
Clock synchronization should be working on the kerberos server [KDC]. The time difference between the client and server machine clocks is configurable in Kerberos 5. A keyword "clockskew" can be set (in seconds), under the [libdefaults] section in [/etc/krb5.conf](#) file, with the value for the difference in time that will be tolerated by the server. The default is 300 seconds [5 minutes]. If the clocks are different by more than 5 minutes, the Kerberos clients will not be able to authenticate to the server. You need to setup a Network Time Protocol compatible client/server network between the three machines. More details on how to setup NTP is available at <http://www.ntp.org/documentation.html>.
To demonstrate the steps, we have used the following names for the systems in the network.

XYZ.ORG	Realm name. The domain the KDC is running for.
kdcserver.xyz.org	The hostname where KDC is running
sshserver.xyz.org	The ssh server hostname
sshclient.xyz.org	The ssh client hostname
krbuser	A user. The principal for this user has to be added to the KDC, and this user must have an account on the sshserver.xyz.org.

The following steps have to be followed to install Kerberos on all the machines, the KDC system, the Server [where SSH server is installed] and the Client machine where SSH client is installed..

The Kerberos installation steps may vary from one Linux distribution to the other. If an rpm is available for Kerberos in the Linux distribution you are using, then install that rpm on all the systems.

Installing Kerberos

Installing Kerberos from source or rpm

The following rpms have to be installed. The versions numbers may differ. Here, the version taken is "1.2.7-10". This is taken on Redhat 9.0.

```
krb5-devel-1.2.7-10.i386.rpm  
krb5-libs-1.2.7-10.i386.rpm  
krb5-workstation-1.2.7-10.i386.rpm
```

The command to install an rpm on Linux is given below.

```
# rpm -ivh <rpm_name>
```

Otherwise, follow these instructions to build Kerberos on your Linux Systems. Download the latest Kerberos 5 source [<http://www.crypto-publish.org/mit-kerberos5/>]

```
Copy the source tar file to /usr/local [You can select other locations also]
You must be a "root" user for performing the following operations
cd /usr/local
tar -xvzf krb5-1.3.1.tar.gz [Ex: We have used 1.3.1 version of Kerberos5 source]
cd krb5-1.3.1
./configure--prefix=/usr/local/krb5
make
make install
```

As explained already, KDC is the main component of Kerberos. The host that runs the KDC will have a name [[Realm](#)] and this consists of a database [list of principals]. The following steps have to be followed for configuring the KDC.

Configuring the KDC

- Create a directory krb5kdc in /usr/local/krb5/var
- Copy the kdc.conf file from /usr/local/krb5-1.3.1/src/config-files to /usr/local/krb5/var/krb5kdc
- Modify the kdc.conf file for the correct information
- Create an access control list file kadm5.acl and save the file in /usr/local/krb5/var/krb5kdc

Sample configuration files are given here. The kdc.conf and kadm5.acl files must be present in the KDC server. The other file krb5.conf must be present on all the machines. These files reflect the details about the realm, the domain-realm mappings and the access privileges for the principals to access the Kerberos database. To setup your own realm and domain, just replace the XYZ.ORG in all the files with your domain name. By convention, all realm names should be in uppercase letters and all DNS hostnames and domain names should be in lower case letters.

KDC Configuration file : kdc.conf

```
# /etc/kdc.conf - Kerberos Key Distribution Center - Configuration File

[kdcdefaults]
kdc_ports = 88

[realms]
XYZ.ORG = {
kadmind_port = 749
max_life = 10h 0m 0s
max_renewable_life = 7d 0h 0m 0s
admin_keytab=/usr/local/krb5/var/krb5kdc/kadm5.keytab
acl_file=/usr/local/krb5/var/krb5kdc/kadm5.acl
master_key_type = des3-hmac-sha1
supported_encetypes = des3-hmac-sha1:normal des-cbc-crc:normal des-cbc-crc:v4
kdc_supported_encetypes = des3-hmac-sha1:normal des-cbc-crc:normal des-cbc-crc:v4
}

[logging]
kdc = FILE:/usr/local/krb5/var/krb5kdc/kdc.log
```

```
admin_server = FILE:/usr/local/krb5/var/krb5kdc/kadmin.log
```

ACL - Access Control List : kadm5.acl

```
# /usr/local/var/krb5/krb5kdc/kadm5.acl - Access Control List configuration File  
*/admin@XYZ.ORG *
```

/etc/krb5.conf

```
# /etc/krb5.conf - Kerberos client Configuration File
```

[logging]

```
default = FILE:/usr/local/krb5/var/log/krb5lib.log  
kdc = FILE:/usr/local/krb5/var/log/krb5kdc.log  
admin_server = FILE:/usr/local/krb5/var/log/kadmin.log
```

[libdefaults]

```
ticket_lifetime = 24000  
default_realm = XYZ.ORG  
dns_lookup_realm = false  
dns_lookup_kdc = false
```

[realms]

```
XYZ.ORG = {  
kdc = kdcserver.xyz.org:88  
admin_server = kdcserver.xyz.org:749  
default_domain = xyz.org  
}
```

[domain_realm]

```
.xyz.org = XYZ.ORG  
xyz.org = XYZ.ORG
```

[kdc]

```
profile = /usr/local/krb5/var/krb5kdc/kdc.conf
```

[pam]

```
debug = false  
ticket_lifetime = 36000  
renew_lifetime = 36000  
forwardable = true  
krb4_convert = false
```

Managing KDC database [Creating KDC database, Adding user principals]

1. First a database has to be created in the KDC host to hold the principals of the users and hosts. The command given here will create the database files. These files will be created in /usr/local/krb5/var/krb5kdc/.

```
# /usr/local/krb5/sbin/kdb5_util create -r XYZ.ORG -s
```

2. This realm [KDC host] needs an administrator. Thus, an entry has to be added in the database for the administrator. The following command do that job.

```
# /usr/local/krb5/sbin/kadmin.local -q "addprinc useradmin/admin@XYZ.ORG"
```

```
Passwd: * * * * *
```

3. The next step is to create a keytab for kadmin [the Kerberos administration server]. This keytab is a file where user keys will be stored and this key will be used by the kadmin to decrypt the administrators Kerberos tickets to determine whether or not it should give the database. This is done with the following command.

```
# /usr/local/krb5/sbin/kadmin.local -q \
```

```
    "ktadd -k /usr/local/var/krb5kdc/kadm5.keytab =>kadmin/admin  
kadmin/changepw"
```

4. For every user in the network, if the user wants to connect to another machine with Kerberos authentication, then, the user must have a principal in the KDC. The following commands are used to add a principal for the user in the KDC database. There must be an user account in the server to which the user wants to connect with this Kerberos principal. For conveniencies, it is recommended to have same names for the user account in the server and the principal name in KDC. [See [Principal](#)].

```
# /usr/local/krb5/sbin/kadmin.local -q "addprinc krbuser"
```

5. Thus, the KDC is configured with a user principal for the user "krbuser". Once this is done, the Kerberos server has to be started.

```
# /usr/local/krb5/sbin/krb5kdc
```

```
# /usr/local/krb5/sbin/kadmin
```

Configuring the Application Server System

The Application server need a keytab file, called /etc/krb5.keytab, to authenticate to the KDC. In order to generate a keytab for a host, the host must have a principal in the Kerberos database. Follow the procedure for adding hosts to the database.

Application server [SSH Server machine]

1. Get the ticket from the KDC for the administrator principal. This command will result in a prompt for the admin password. Give a password for the admin.

```
# /usr/local/krb5/bin/kinit admin/admin
```

```
Passwd: * * * * *
```

2. Now, the principal of the server [sshserver.xyz.org] must be added to the KDC.

```
# /usr/local/krb5/sbin/kadmin.local -q "addprinc -randkey host/sshserver.xyz.org"
```

3. Create a keytab file. This will create a keytab file in /etc/krb5.keytab.

```
# /usr/local/krb5/sbin/kadmin.local -q " ktadd -k /etc/krb5.keytab  
host/sshserver.xyz.org"
```

```

4. Check the correct information in the keytab file. ktutil is an utility for managing
the keylist in keytab files. "rkt" means read keytab. "l" lists the current keylist in
the keytab file. To know the correct server principal entry in the keytab file, ktutil
command is used. "q" is used to quit from the kutil service.
# /usr/local/krb5/sbin/ktutil
ktutil: rkt /etc/krb5.keytab
ktutil: l
slot KVNO Principal
-----
1 3 host/sshserver.xyz.org@XYZ.ORG
2 3 host/sshserver.xyz.org@XYZ.ORG
ktutil: q

```

Configuring the Application Client System

Client machine needs the programs "kinit", "klist", "kdestroy" and the /etc/krb5.conf file which contains the KDC server details. The installation of kerberos-1.3.1 creates all the client programs in /usr/local/krb5/bin

OpenSSH Authentications

Kerberos5 server can be used for the following authentications with OpenSSH.

- GSS-API Authentication
- Kerberos5 Authentication

OpenSSH must be available on the client and server machines. [on sshclient.xyz.org and sshserver.xyz.org respectively]. For more details about OpenSSH build and installation, see [OpenSSH installation on Linux.](#)

GSS-API Authentication

1. Edit the ssh server configuration file [by default, in /etc/ssh/sshd_config] on sshserver.xyz.org and ssh client configuration file [by default, in /etc/ssh/ssh_config] on sshclient.xyz.org. If you had given a different installation path for the OpenSSH application, then check sshd_config in "etc" directory in the installed path.

GSSAPIAuthentication yes

2. Start the sshd server on sshserver.xyz.org. By default the ssh server will run on port 22.

```
# /user/sbin/sshd
```

3. Make sure that the kdc server is running in the KDC machine , [kdcserver.xyz.org]

```
# ps -A |grep krb5kdc
```

If the server is not started, start by

```
# /usr/local/krb5-1.3.1/sbin/krb5kdc
```

```
# /usr/local/krb5-1.3.1/sbin/kadmind
```

4. Get a initial ticket ([TGT](#)) in the client machine [sshclient.xyz.org]
kinit krbuser
Passwd: * * * * *

5. Once a TGT is obtained, it can be checked as follows.
klist

Ticket cache: FILE:/tmp/krb5cc_11190
Default principal: krbuser@XYZ.ORG

Valid starting Expires Service principal
01/09/04 12:41:09 01/09/04 22:41:09 [krbtgt/XYZ.ORG@XYZ.ORG](#)

6. Connect the ssh client with the server
ssh krbuser@sshserver.xyz.org

Note:
GSS-API authentication is a password less authentication. For a successful connection, the server should not ask for a password for the authentication. But, other authentications such as Publickey will also connect without password if there is proper key on the client and server. To make sure that GSS-API authentication is succeeded, the ssh client can be run in verbose mode. This can be done by using "-v" with the ssh client. The more the number of "v", the more the verbose details. The maximum number of "v" that can be given is 3.

ssh krbuser@sshserver.xyz.org -vvv
.....
debug1: Authentication succeeded (gssapi)

Kerberos Authentication

Kerberos5 Authentication

1. Edit the ssh server configuration file [by default, in /etc/ssh/sshd_config] on sshserver.xyz.org and ssh client configuration file [by default, in /etc/ssh/ssh_config] on sshclient.xyz.org. If you had given a different installation path for the OpenSSH application, then check sshd_config in "etc" directory in the installed path.

KerberosAuthentication yes

2. Start the sshd server on sshserver.xyz.org. By default the ssh server will run on port 22.

/usr/sbin/sshd

3. Make sure that the kdc server is running in the KDC machine , [kdcserver.xyz.org]

ps -A |grep krb5kdc

If the server is not started, start by

/usr/local/krb5-1.3.1/sbin/krb5kdc

/usr/local/krb5-1.3.1/sbin/kadmind

	<p>4. Destroy the ticket in the client machine # kdestroy</p> <p>5. Connect the ssh client with the server. # ssh krbuser@sshserver.xyz.org</p> <p>Note: In this authentication, the user will be prompted for a password. The password to be entered should be the "Kerberos password". This is the password that is used when "kinit" is done.</p>
--	--

RFCs for Kerberos	<ul style="list-style-type: none">• RFC 1510 The Kerberos Network Authentication Service [V5]• RFC 1964 The Kerberos Version 5 GSS-API mechanism• RFC 1305 Network Time Protocol
--------------------------	--

References	<ul style="list-style-type: none">• http://web.mit.edu/kerberos/www/ - Kerberos home page on MIT• http://web.mit.edu/kerberos/www/dialogue.html - A good explanation of Kerberos protocol, in plain english. The conversation style is intended to introduce the concept for people who are unfamiliar with Kerberos. Technical details are kept minimum.• http://www.cs.uu.nl/wais/html/na-dir/kerberos-faq/general.html - Kerberos FAQs• Book - "SSH, The Secure Shell: The Definitive Guide". Authors : Daniel J. Barrett and Richard E. Silverman.
-------------------	---

Support	If you still have problems in configuring Kerberos or OpenSSH, you can contact ViSolve OpenSSH Support team at support@visolve.com for support.
----------------	--

About ViSolve.com <p>ViSolve is an international corporation that provides technical services, for Internet based systems, for clients around the globe. ViSolve is in the business of providing software solutions since 1995. We have experience of executing several major projects and we are now completely focused on leading Internet technologies, Testing QA and support. We are committed to the Open source movement and in the same lines we provide free support for products like Linux, Apache and Squid to the user community.</p>
--

Document Version : 1.0 Created On : 28-01-02 Updated On : 29-05-06
--