Configuring a Kerberos 5 Server

When setting up Kerberos, install the server first. If it is necessary to set up slave servers, the details of setting up relationships between master and slave servers are covered in the *Kerberos 5 Installation Guide* located in the `/usr/share/doc/krb5-server-<version-number>` directory (replace *<version-number>* with the version number of the `krb5-server` package installed on the system).

To configure a basic Kerberos server, follow these steps:

1. Be sure that clock synchronization and DNS are functioning on all client and server machines before configuring Kerberos 5. Pay particular attention to time synchronization between the Kerberos server and its clients. If the server and client clocks are different by more than five minutes (this default amount is configurable in Kerberos 5), Kerberos clients can not authenticate to the server. This clock synchronization is necessary to prevent an attacker from using an old Kerberos ticket to masquerade as a valid user.

   It is advisable to set up a Network Time Protocol (NTP) compatible client/server network even if Kerberos is not being used. Red Hat Enterprise Linux includes the `ntp` package for this purpose. Refer to `/usr/share/doc/ntp-<version-number>/index.htm` for details about how to set up Network Time Protocol servers and http://www.eecis.udel.edu/~ntp for additional information about NTP.

2. Install the `krb5-libs`, `krb5-server`, and `krb5-workstation` packages on the dedicated machine which runs the KDC. This machine needs to be very secure — if possible, it should not run any services other than the KDC.

   If a graphical user interface is required to administrate Kerberos, install the `gnome-kerberos` package. It contains `krb5`, a GUI tool for managing tickets.

3. Edit the `/etc/krb5.conf` and `/var/kerberos/krb5kdc/kdc.conf` configuration files to reflect the realm name and domain-to-realm mappings. A simple realm can be constructed by replacing instances of *EXAMPLE.COM* and *example.com* with the correct domain name — being certain to keep uppercase and lowercase names in the correct format — and by changing the KDC from *kerberos.example.com* to the name of the Kerberos server. By convention, all realm names are uppercase and all DNS hostnames and domain names are lowercase. For full details about the formats of these configuration files, refer to their respective man pages.

4. Create the database using the `kdb5_util` utility from a shell prompt:

   ```
   /usr/kerberos/sbin/kdb5_util create -s
   ```

   The `create` command creates the database that stores keys for the Kerberos realm. The `-s` switch forces creation of a *stash* file in which the master server key is stored. If no stash file is present from which to read the key, the Kerberos server (`krb5kdc`) prompts the user for the master server password (which can be used to regenerate the key) every time it starts.

5. Edit the `/var/kerberos/krb5kdc/kadm5.acl` file. This file is used by `kadmind` to determine which principals have administrative access to the Kerberos database and their level of access. Most organizations can get by with a single line:

   ```
   */admin@EXAMPLE.COMï¿½ï¿½*
   ```

   Most users are represented in the database by a single principal (with a *NULL*, or empty, instance, such as *joe@EXAMPLE.COM*). In this configuration, users with a second principal with an instance of *admin* (for example, *joe/admin@EXAMPLE.COM*) are able to wield full power over the realm's Kerberos database.

   Once `kadmind` is started on the server, any user can access its services by running `kadmin` on any of the clients or servers in the realm. However, only users listed in the `kadm5.acl` file can modify the database in any way, except for changing their own passwords.

   

   Note

   The `kadmin` utility communicates with the `kadmind` server over the network, and uses Kerberos to handle authentication. For this reason, the first principal must already exist before connecting to the server over the network to administer it. Create the first principal with the `kadmin.local` command, which is specifically designed to be used on the same host as the KDC and does not

use Kerberos for authentication.

Type the following `kadmin.local` command at the KDC terminal to create the first principal:

```
/usr/kerberos/sbin/kadmin.local -q "addprinc username/admin"
```

6. Start Kerberos using the following commands:

```
/sbin/service krb5kdc start
/sbin/service kadmin start
/sbin/service krb524 start
```

7. Add principals for the users using the `addprinc` command with `kadmin`. `kadmin` and `kadmin.local` are command line interfaces to the KDC. As such, many commands are available after launching the `kadmin` program. Refer to the `kadmin` man page for more information.

8. Verify that the KDC is issuing tickets. First, run `kinit` to obtain a ticket and store it in a credential cache file. Next, use `klist` to view the list of credentials in the cache and use `kdestroy` to destroy the cache and the credentials it contains.

Note

By default, `kinit` attempts to authenticate using the same system login username (not the Kerberos server). If that username does not correspond to a principal in the Kerberos database, `kinit` issues an error message. If that happens, supply `kinit` with the name of the correct principal as an argument on the command line (`kinit <principal>`).

Once these steps are completed, the Kerberos server should be up and running.

Configuring a Kerberos 5 Client

Setting up a Kerberos 5 client is less involved than setting up a server. At a minimum, install the client packages and provide each client with a valid `krb5.conf` configuration file. Kerberized versions of `rsh` and `rlogin` also requires some configuration changes.

1. Be sure that time synchronization is in place between the Kerberos client and the KDC. Refer to Section 19.5 *Configuring a Kerberos 5 Server* for more information. In addition, verify that DNS is working properly on the Kerberos client before configuring the Kerberos client programs.
2. Install the `krb5-libs` and `krb5-workstation` packages on all of the client machines. Supply a valid `/etc/krb5.conf` file for each client (usually this can be the same `krb5.conf` file used by the KDC).
3. Before a workstation in the realm can allow users to connect using kerberized `rsh` and `rlogin`, that workstation must have the `xinetd` package installed and have its own host principal in the Kerberos database. The `kshd` and `klogind` server programs also need access to the keys for their service's principal.

   Using `kadmin`, add a host principal for the workstation on the KDC. The instance in this case is the hostname of the workstation. Use the `-randkey` option for the `kadmin`'s `addprinc` command to create the principal and assign it a random key:

   ```
   addprinc -randkey host/blah.example.com
   ```

   Now that the principal has been created, keys can be extracted for the workstation by running `kadmin` *on the workstation itself*, and using the `ktadd` command within `kadmin`:

   ```
   ktadd -k /etc/krb5.keytab host/blah.example.com
   ```

4. To use other kerberized network services, they must first be started. Below is a list of some common kerberized services and instructions about enabling them:

   - `rsh` and `rlogin` — To use the kerberized versions of `rsh` and `rlogin`, enable `klogin`, `eklogin`, and `kshell`.
   - Telnet — To use kerberized Telnet, `krb5-telnet` must be enabled.
   - FTP — To provide FTP access, create and extract a key for the principal with a root of `ftp`. Be certain to set the instance to the fully qualified hostname of the FTP server, then enable `gssftp`.
   - IMAP — To use a kerberized IMAP server, the `cyrus-imap` package uses Kerberos 5 if it also has the `cyrus-sasl-gssapi` package installed. The `cyrus-sasl-gssapi` package contains the Cyrus SASL plugins which support GSS-API authentication. Cyrus IMAP should function properly with Kerberos as long as the `cyrus` user is able to find the proper key in `/etc/krb5.keytab`, and the root for the principal is set to `imap` (created with `kadmin`).

     The `dovecot` package also contains an IMAP server alternative to `cyrus-imap`, which is also included with Red Hat Enterprise Linux, but does not support GSS-API and Kerberos to date.
   - CVS — To use a kerberized CVS server, `gserver` uses a principal with a root of `cvs` and is otherwise identical to the CVS `pserver`.

   For details about how to enable services, refer to the chapter titled *Controlling Access to Services* in the *Red Hat Enterprise Linux System Administration Guide*.