Install a Trust Router on RHEL/CentOS/SL 6

On this page you will find instructions on how to set up a Trust Router on RedHat, CentOS, or Scientific Linux 6.

Contents

This guide assumes that you are using the latest available version of RHEL/CentOS/SL 6 - which at the time of writing this guide is 6.5.

1. System Preparation

1.1. Install CentOS 6

The first thing that is required is a CentOS machine - this can be physical or virtual.

1. Install the operating system via usual mechanism (e.g., net boot CD, ISO in VMware/VirtualBox or the DVD image).
2. Choose the following server install options: "Basic server".
3. Create/choose a secure root password and an initial system user account.
4. Once installed, make sure you run a `yum makecache` and `yum update` to ensure your system is fully up to date.

Tip

We would recommend using LVM when disk partitioning to allow easier partition/disk expansion on a live system.

Warning

After install, you will want to secure/lockdown the server as best practice dictates - for both the server and any extra software installed. This is beyond the remit of this guide but there are many guides available that provide information on how to secure your CentOS servers and applications.

1.2. Configure CentOS 6

Next, there are a few CentOS configuration options that need to be set in advance.

1.2.1. SELinux configuration

There are a few SELinux policies missing for Moonshot. Hence, SELinux must either be run in Permissive mode, or the `radius` and `radiusd_moonshot` policies be disabled whenever FreeRadius is used (ie. APC, IDPs and RPPs)

For disabling the `radius` and `radiusd_moonshot` policies and let FreeRadius run unconfined, use the following steps:

```
$ semodule -d radiusd_moonshot
$ semodule -d radius
$ restorecon -R -v /usr/sbin/radiusd /var/log/radius/
```

For setting SELinux in Permissive mode, please refer to the RootUsers guide to SELinux. It also applies to CentOS 6: https://www.rootusers.com/how-to-enable-or-disable-selinux-in-centos-rhel-7/

1.2.2. Networking configuration

For production deployments, it is recommended that the machine be assigned a static IP address.

For CentOS networking information please refer to the ServerLab guide for CentOS 6: https://www.serverlab.ca/tutorials/linux/administration-linux/configure-centos-6-network-settings/

1.2.3. Firewall configuration

The following ports are required to be accessible from the outside world, both in the local firewall and in any external firewalls:

- 2083/tcp (for RadSec connections to other Moonshot entities)
- 12309/tcp (for Trust Router client connections - if using the Trust Router to broker trust relationships between entities)

Here are sample firewall rules that establish incoming and outgoing rules to both the Test and Live (Jisc Assent) Moonshot trust router infrastructures. If you connect to another Trust Router, adjust these rules to suit:

IP Tables sample firewall rules (Jisc Assent)

```
-A INPUT -m state --state NEW,ESTABLISHED,RELATED -m tcp -p tcp -s 0/0 --dst <IdP/RP Proxy IP address> --
dport 2083 -j ACCEPT
-A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -m tcp -p tcp -s <IdP/RP Proxy IP address> --dst 0/0 --
dport 2083 -j ACCEPT
 -A INPUT -m state --state NEW,ESTABLISHED,RELATED -m tcp -p tcp -s
212.219.179.130,212.219.179.131,212.219.179.138,212.219.179.146 --dst <IdP/RP Proxy IP address> --dport
12309 -j ACCEPT
-A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -m tcp -p tcp -s <IdP/RP Proxy IP address> --dst
212.219.179.130,212.219.179.131,212.219.179.138,212.219.179.146 --dport 12309 -j ACCEPT
```

IP Tables sample firewall rules (Test Network)

```
-A INPUT -m state --state NEW,ESTABLISHED,RELATED -m tcp -p tcp -s 0/0 --dst <IdP/RP Proxy IP address> --
dport 2083 -j ACCEPT
-A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -m tcp -p tcp -s <IdP/RP Proxy IP address> --dst 0/0 --
dport 2083 -j ACCEPT
-A INPUT -m state --state NEW,ESTABLISHED,RELATED -m tcp -p tcp -s 13.79.134.211,13.79.128.103,52.169.31.1
04 --dst <IdP/RP Proxy IP address> --dport 12309 -j ACCEPT
-A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -m tcp -p tcp -s <IdP/RP Proxy IP address> --dst 13.79.
134.211,13.79.128.103,52.169.31.104 --dport 12309 -j ACCEPT
```

1.3. Add the Required Repositories

Moonshot requires two `yum` repositories to be added to the system - EPEL (home of some required dependencies), and the Moonshot repository itself.

1. Install EPEL by running the following command:

   ```
   $ yum install epel-release
   ```

   Depending on your platform, the `epel-release` package is part of one of the optional repositories. On CentOS, it is part of the Extras repository. On RHEL, you must enable both the Optional and Extras repositories. For more information, visit the EPEL homepage.

2. Install the Moonshot repository by creating a new file at `/etc/yum.repos.d/moonshot.repo` with the following content:

   ```
   [Moonshot]
   name=Moonshot
   baseurl=http://repository.project-moonshot.org/rpms/centos6/
   failovermethod=priority
   gpgcheck=1
   gpgkey=file:///etc/pki/rpm-gpg/Moonshot
   ```

3. Install the Moonshot GPG key:

   ```
   $ wget -O /etc/pki/rpm-gpg/Moonshot http://repository.project-moonshot.org/rpms/centos6/moonshot.key
   ```

> **Verifying the Moonshot GPG key**
>
> If you wish to verify the Moonshot GPG key's validity and integrity, please see the Packaging GPG Key for further details.

4. Install the official Shibboleth repository.

```
$ wget -O /etc/yum.repos.d/shibboleth.repo http://download.opensuse.org/repositories/security:/shibboleth
/CentOS_CentOS-6/security:shibboleth.repo
```

2. Install Trust Router

We're now ready to install the Trust Router software and its required dependencies. Install the software by running the following command:

```
$ yum install trust_router moonshot-ui
```

3. Configure Trust Router

Next, we need to configure the Trust Router.

3.1. RadSec

3.1.1. APC TLS

First, you will need a copy of a client key and certificate (and appropriate CA) from the APC(s) that your Trust Router serves. Copy them onto the filesystem.

> You can put these files anywhere on the file system, but this guide assumes you put them in `/etc/pki/tls`. If you place them in a different location you will need to change the locations below as appropriate.

3.1.2. Connection to APC

Next, we need to configure the RadSec configuration for the APC. We do this by creating a file at `/etc/radsec.conf` with the following:

```
realm gss-eap {
        type = "TLS"
        cacertfile = "/etc/pki/tls/tr-ca.pem"
        certfile = "/etc/pki/tls/tr-client.pem"
        certkeyfile = "/etc/pki/tls/tr-client.key"
        disable_hostname_check = yes
        server {
                hostname = "YOUR_APC_HOSTNAME"
                service = "2083"
                secret = "radsec"
        }
}
```

Then check the file and the certificates can be read by the Trust Router user:

> There is currently an error in the home directory for the trustrouter user in `/etc/passwd`.
>
> Edit the file and change the home directory from `/var/lib/trustrouter` to `/var/lib/trust_router` before executing the below commands.

```
$ sudo su - --shell=/bin/bash trustrouter
$ cat /etc/radsec.conf
$ cat /etc/pki/tls/tr-*.*
```

3.2. Trust Router

3.2.1. Daemon Configuration

Your Trust Router will need to have a few core configuration items set. To do this:

1. Open the default instance's main configuration file at `/etc/trust_router/conf.d/default/main.cfg` for editing.
    a. Change the hostname to the (fully qualified) hostname of your Trust Router.
    b. Change the port that it runs on, if necessary.
2. Open the `/etc/sysconfig/trust_router` file for editing. Make sure the configuration items are correct. Items you will most likely have to change are:
    - TR_DEFAULT_TEST_ACCEPTOR - this will need to be set to the (fully qualified) hostname of your Trust Router.
    - TR_DEFAULT_TEST_RPREALM - this will need to be set to the (fully qualified) hostname for your APC.
    - TR_DEFAULT_TEST_COMMUNITY - this will need to be set to the (fully qualified) hostname for your APC.
    - TR_DEFAULT_TEST_REALM - this will need to be set to the (fully qualified) hostname for your APC.

3.2.2. Moonshot Configuration

Moonshot, you say? Yes, Trust Router uses Moonshot to authenticate and secure all communications between Trust Router clients and servers. So, you will need to configure the trust router user to make use of the Moonshot flatstore (i.e. telling Moonshot that this is a special system account, not a regular user account), and you will need to import a set of credentials for your Trust Router to use.

1. Enable the trustrouter user to use the Moonshot UI flatstore:

```
$ echo "trustrouter" >> /etc/moonshot/flatstore-users
```

2. Import it using the `moonshot-webp` command (as the trustrouter user):

> ⚠ There is currently an error in the home directory for the trustrouter user in `/etc/passwd`.
>
> Edit the file and change the home directory from `/var/lib/trustrouter` to `/var/lib/trust_router` before executing the below commands.

```
$ su - --shell=/bin/bash trustrouter
$ unset DISPLAY
$ moonshot-webp -f [path to credential file]
```

> ⓘ The credentials file will be given to you by the administrator of the APC.

3.2.3. Shibboleth

Shibboleth, you say? Yes, Shibboleth is used by the Moonshot components to be able to deal with incoming SAML. However, this feature typically isn't used in Trust Router, but its logging will appear in your Trust Router's log files. So, to simplify your log files, it is recommended that you silence the Shibboleth logging. To do this:

1. Open `/etc/shibboleth/console.logger` for editing.
2. Change WARN to NONE on the first line, i.e.

```
log4j.rootCategory=NONE, console
```

3.2.4. Default Peer

> ⓘ If your Trust Router is going to run in its own, standalone, trust network, then skip this step.

If your Trust Router is going to run in a wider trust network, then you can configure your Trust Router's default peer - i.e. the Trust Router it sends its clients to when they ask it to locate a Moonshot entity that your Trust Router doesn't know about. To do this:

1. Open `/etc/trust_router/conf.d/default/peering.cfg` for editing. Change the content as follows:

```
{
"default_servers": [
 "[hostname of trust router]"
 ]
}
```

If the `/etc/trust_router` directory does not exist, you may need to create it yourself, along with the subdirectories mentioned.

Example

If you were configuring your default Trust Router peer to be Janet's Trust Router at tr.moonshot.ja.net, its `peering.cfg` file would look like this:

```
{
"default_servers": [
 "tr.moonshot.ja.net"
 ]
}
```

3.2.5. Configure your Trust Router

A trust router requires a trust configuration to function correctly. See the trust configuration file for more information.

Place an appropriate `trusts.cfg` file info the `/etc/trust_router` directory and symbolically link it into the default configuration directory:

```
# cd /etc/trust_router/conf.d/default
# ln -s ../../trusts.cfg
```

You can find a Trust Router configuration suitable for a Trust Router connecting to tr.moonshot.ja.net at sample Trust Router Client configuration

3.2.6. Start your Trust Router

You are now ready to restart your Trust Router and test it. To do this:

1. As root, start the Trust Router daemon:

```
$ service trust_router start
```

4. Testing

To test your trust router, you should attempt a TIDC request on a Moonshot service connected to your trust router. If you have defined a default peer, the TIDC request may take a little longer, but it should succeed.

If it fails, please contact us.

5. Next Steps

At this point, you now have a Trust Router.

5.1. Automatically start the software

5.1.1. Trust Router

To automatically start Trust Router, issue the following command (as root):

```
$ sudo chkconfig trust_router on
$ sudo service trust_router restart
```

If this is working correctly, you should see trust_router running as a daemon process.