Configure the Moonshot RP Proxy to Talk to Applications /Services

Once you have a Moonshot RP Proxy installed and configured, any new application or service that wishes to make use of Moonshot authentication will need to be configured within your Moonshot RP Proxy as a registered client.

Contents

1. Introduction

Any application or service that wishes to use a Moonshot RP Proxy as its gateway to the world of Moonshot must be configured as a registered client within the RP Proxy's configuration.

2. Adding a new Client

FreeRADIUS clients are normally added into FreeRADIUS' clients.conf. However, Moonshot services are a special type of FreeRADIUS client and it is recommended to configure the connection to them over TLS; this needs to be configured in the abfab-tls file.

The location of FreeRADIUS' abfab-tls file will differ depending on the distribution that you installed it on.

- On Debian, this will be /etc/freeradius/sites-available/abfab-tls
- On RHEL/CentOS/SL, this will be /etc/raddb/sites-available/abfab-tls

This assumes you installed as a package - if you built it by hand, it'll be wherever you configured it to live.

FreeRADIUS' clients.conf file has many examples of different ways of defining a client. This page will just give a few simple options; consult the examples or the FreeRADIUS documentation for further details.

2.1. Using Certificates

To add a new client using a simple shared secret, open FreeRADIUS's abfab-tls file for editing. At the bottom of the file is a block labelled "clients radsec-abfab {}". All Moonshot clients need to be added in here.

```
client example-client {
        ipaddr                  = {IP address of server}
    proto                   = tls
    shortname               = {a short name for the client}
    secret                  = radsec
    gss_acceptor_host_name  = {fqdn of the client}
    gss_acceptor_realm_name = {realm used by the client
    trust_router_coi        = {COI that the service belong to}
}
```

Example

If we were going to add a client for an SSH server whose hostname was ssh.example.com, registered with a Trust Router with a realm of example.com, you would have the following in the "clients radsec-abfab" section of abfab-tls:

```
client example-client {
        ipaddr                  = 192.168.0.1
    proto                   = tls
    shortname               = example
    secret                  = radsec
    gss_acceptor_host_name  = ssh.example.com
    gss_acceptor_realm_name = example.com
    trust_router_coi        = apc.moonshot.ja.net
}
```

3. Configuring the Client

Now that your Moonshot RP Proxy is configured to recognise and communicate with a particular set of clients, those clients need to be configured to talk to it. See the Configure a Linux Server to Connect to an RP Proxy page for details on how to do this.

3.1. Generating Client Certificates

As the Moonshot RP Proxy administrator, if using TLS as recommended, you will need to give the administrator of the client that wants to connect to your RP a client certificate to be able to do so. To do this:

3.1.1. Edit the client.cnf file

In FreeRADIUS' certs directory (`/etc/raddb/certs/`) is a file named "client.cnf". This is the source of information used to generate a new client certificate. You should edit the [client] section of this file, changing the emailAddress and CommonName fields as appropriate.

Generate the new certificate

3.1.2. Generate the new certificate

In the same directory, run the following command:

```
$ make client
```

3.1.3. Give files to client

Finally, you should now give the adminstrator of the client the following files (which will all be located in FreeRADIUS' certs directory:

- ca.pem
- client.pem
- client.key