_CertPrep_RHEL

Certificates

We need to get FreeRADIUS to create some private and public keys to use for its RadSec connections. Create and install the certificates by doing the following (as root).

1. Change into the `/etc/raddb/certs` directory

```
cd /etc/raddb/certs
```

2. Edit the certificate generation properties in *client.cnf*, *server.cnf*, and *ca.cnf* as follows:

   a. In the `ca.cnf` file:
      i. In the `[ req ]` section, add `encrypt_key = no`
      ii. In the `[CA_default]` section, change the default_days from 60 to a higher number (this is how long the certificates you create will be valid for). When the certificates expire, you will have to recreate them.
      iii. in the `[ certificate_authority ]` section, change all of the parameters to match those of your organisation. e.g.

```
[certificate_authority]
countryName             = GB
stateOrProvinceName     = England
localityName            = Camford
organizationName        = Camford University
emailAddress            = support@camford.ac.uk
commonName              = "Camford University FR Certificate Authority"
```

   b. In the `server.cnf` file:

      i. In the `[ req ]` section, add `encrypt_key = no`
      ii. In the `[CA_default]` section, change the default_days from 60 to a higher number (this is how long the certificates you create will be valid for). When the certificates expire, you will have to recreate them.
      iii. in the `[ server ]` section, change all of the parameters to match those of your organisation. e.g.

```
[server]
countryName             = GB
stateOrProvinceName     = England
localityName            = Camford
organizationName        = Camford University
emailAddress            = support@camford.ac.uk
commonName              = "Camford University FR Server Certificate"
```

> When changing passwords in the `[ req ]` section of the server.cnf file, you must also update the `private_key_password` option in the FreeRADIUS `mods-available/eap` file with the same password.
>
> We recommend that you do **not** change these defaults.

   c. In the `client.cnf` file:

      i. In the `[ req ]` section, add `encrypt_key = no`
      ii. In the `[CA_default]` section, change the default_days from 60 to a higher number (this is how long the certificates you create will be valid for). When the certificates expire, you will have to recreate them.
      iii. in the `[ client ]` section, change all of the parameters to match those of your organisation. e.g.

```
[client]
countryName             = GB
stateOrProvinceName     = England
localityName            = Camford
organizationName        = Camford University
emailAddress            = support@camford.ac.uk
commonName              = "Camford University FR Client Certificate"
```

> All of the organisation parameters (`countryName`, `localityName`, etc) need to match in the three .cnf files but the `commonName` must be unique in each file)

3. Clear out any old certificates in the directory:

```
make destroycerts
```

4. Run the bootstrap script to generate the certificates

```
./bootstrap
```

5. Create a file that is the concatenation of the certificate and private key of the client.

```
openssl x509 -in client.crt > client.pem ; cat client.key >> client.pem
```

6. Because the above command was run as root, the keys and certificates created will not be readable by the FreeRADIUS user by default, and FreeRADIUS will not be able to start. To fix this, reset the group for the files:

```
chgrp radiusd {client,server,ca,dh}*
```

OpenSSL settings (CentOS 6 only)

By default, FreeRADIUS attempts to detect the version of OpenSSL that is installed to block vulnerable versions. However, RedHat/CentOS/Scientific Linux patch existing versions, which may lead FreeRADIUS to believe that the installed version is unsafe. This setting overrides the check.

1. Open `/etc/raddb/radiusd.conf` for editing:
    a. Search for the `allow_vulnerable_openssl` setting in the security { } section.
    b. Edit it like so:

```
        # allow_vulnerable_openssl = no
        allow_vulnerable_openssl = 'CVE-2016-6304'
```

RadSec

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

```
realm gss-eap {
        type = "TLS"
        cacertfile = "/etc/raddb/certs/ca.pem"
        certfile = "/etc/raddb/certs/client.pem"
        certkeyfile = "/etc/raddb/certs/client.key"
        disable_hostname_check = yes
        server {
                hostname = "127.0.0.1"
                service = "2083"
                secret = "radsec"
        }
}
```

Dynamic Realm support

We need to tell your FreeRADIUS server to support dynamic lookup of realms.

1. Open `/etc/raddb/proxy.conf` for editing:
    a. Towards the top of the file is a stanza beginning "`proxy server {`". Find this.
    b. Below this, add `dynamic = yes`, like so:

```
proxy server {
        dynamic = yes
```