Using TLS with Radmind

Version 1.4.0rc1

This document describes how to configure and use certificates for the radmind client and server.

Creating a Certificate Authority

1. Create the Certificate Authority directory structure:

```
[server] root# cd /var/radmind
[server] root# mkdir CA
[server] root# mkdir CA/certs
[server] root# mkdir CA/crl
[server] root# mkdir CA/newcerts
[server] root# mkdir CA/private
[server] root# echo "01" > CA/serial
[server] root# touch CA/index.txt
```

2. Download the example OpenSSL Configuration File from

```
[server] root# cd /var/radmind/CA
[server] root# curl -O –L http://radmind.org/files/openssl.cnf
```

3. Create a self-signed certificate authority (CA) certificate and an encrypted private key.

```
[server] root# cd /var/radmind/CA
[server] root# openssl req -new -x509 –days 400 -keyout \
               private/CAkey.pem -out ca.pem -config openssl.cnf
```

When prompted, enter a PEM pass phrase, common name and email address for the CAkey. The common name should indicate the CA you are creating, such as radmind-ca. The email address should be the administrative contact for you CA.

Creating a Certificate for the radmind server

1. Create a certificate request and an unencrypted private key. This certificate will only be valid for 360 days. A certificate will no longer work once it has expired.

```
[server] root# cd /var/radmind/CA
[server] root# openssl req -new -keyout key.pem -out req.pem \
               -days 360 -config openssl.cnf -nodes
```

When prompted, enter the common name and email address for the radmind server certificate. The common name should be the fully qualified domain name of your radmind server. The email address should be the administrative contact for you radmind server.

2. Sign the certificate request with the CA's certificate and private key.

```
[server] root# cat req.pem key.pem > new-req.pem
[server] root# openssl ca -policy policy_match -out out.pem \
               -config openssl.cnf -infiles new-req.pem
```

When prompted, enter the pass phase for CAkey, and confirm the signing and commit the certificate.

3. Combine the certificate and key into one file:

```
[server] root# cat out.pem key.pem > cert.pem
```

4. Remove temporary files

```
[server] root# rm req.pem new-req.pem out.pem
```

5. Install the server and ca certificates

```
[server] root# cp cert.pem /var/radmind/cert
[server] root# cp ca.pem /var/radmind/cert
```

Creating a Certificate for a radmind client

1. Create a certificate request and an unencrypted private key. This certificate will only be valid for 360 days. A certificate will no longer work once it has expired.

```
[server] root# cd /var/radmind/CA
[server] root# openssl req -new -keyout key.pem -out req.pem \
               -days 360 -config openssl.cnf -nodes
```

When prompted, enter the common name and email address for the radmind server certificate. The common name should be what you will list in the radmind config file. Each client that uses this certificate will be assigned the same command file by the radmind server. The email address should be the administrative contact for this client server.

2. Sign the certificate request with the CA's certificate and private key.

```
[server] root# cat req.pem key.pem > new-req.pem
[server] root# openssl ca -policy policy_match -out out.pem \
               -config openssl.cnf -infiles new-req.pem
```

When prompted, enter the pass phase for CAkey, and confirm the signing and commit the certificate.

3. Combine the certificate and key into one file:

```
[server] root# cat out.pem key.pem > client-cert.pem
```

4. Remove temporary files

```
[server] root# rm req.pem new-req.pem out.pem
```

TLS & radmind

With TLS, radmind is able to create an encrypted channel on which to communicate, and depending on the level of TLS implemented, verify the client and server. Each radmind environment will need a single certificate authority and minimally a certificate for the server. If you want to verify the client, you will also need to create a client certificate.

Authorization level 0 – No TLS

At the level, TLS is not used. This is the default level.

Authorization level 1 – Server Verification

At this level, the connection between the radmind server and client is encrypted. The client is also able to verify the server. To implement this level, follow these steps:

1. Create a certificate authority on the radmind server
2. Create a certificate for the radmind server.
3. Added the CA's certificate to /var/radmind/cert on the client
4. Add –w 1 as an option to the radmind server startup script or set "SSL Authorization/Encryption" to "Verify Server" in the Radmind Server Preferences on the client and server.
5. Restart the server.

To use authorization level 1, add `–w 1` as command line option to each tool that connects with the server.

Authorization level 2 – Client and Server Verification

At this level, the connection between the radmind server and client is encrypted. The client and server also verify each other. To implement this level, follow these steps:

1. Create a certificate authority on the radmind server
2. Create a certificate for the radmind server.
3. Create a certificate for the client.
4. Copy the client's certificate into /var/radmind/cert/cert.pem on the client
5. Added the CA's certificate to /var/radmind/cert/ca.pem on the client
6. Add –w 2 as an option to the radmind server startup script or set "SSL Authorization/Encryption" to "Verify Client & Server" in the Radmind Server Preferences on the client and server.
7. Restart the server.

To use authorization level 2, add `–w 2` as command line option to each tool that connects with the server.

More Information

- Openssl man pages
- http://www.pseudonym.org/ssl/ssl_cook.html

Contributers

- Patrick McNeal, University of Michigan
- Rebecca Darling, Wellesley College