Open ssl

OpenSSL enables encrypted communication between the client and the server. For ecFlow, this can be used for user commands.

To enable this for ecflow 4, please ensure you build ecFlow with '-DENABLE_SSL'. You will need to ensure that open SSL is installed on your system.

This is **enabled by default for ecflow 5 if** the SSL libraries are found on the system.

Check if openssl enabled for ecflow

```
ecflow_client --version # look for a string openssl
ecflow_server --version # look for a string openssl
```

In order to use OpenSSL, we need to set up some certificates. (These will be self-signed certificates, rather than a certificate authority).

The ecFlow client and server will look for the certificates in **$HOME/.ecflowrc/ssl** directory.

ecFlow server expects the following files in : **$HOME/.ecflowrc/ssl**

- dh2048.pem
- server.crt
- server.key
- server.passwd (optional) if this exists it must contain the passphrase used to create server.key.

ecFlow client expects the following files in: **$HOME/.ecflowrc/ssl**

- server.crt (this must be the same as the server)

The following steps, show you how to create these files:

- Generate a password-protected private key. This will request a passphrase.

  This key is a 1024 bit RSA key which is encrypted using Triple-DES and stored in a PEM format so that it is readable as ASCII text

  Password protected private key

  ```
  openssl genrsa -des3 -out server.key 1024
  ```

- If you want additional security. Create a file called **'server.passwd'** and add the passphrase to the file. Then set the file permission so that the file is only readable by the server process.

  Or you can choose to remove the password requirement. In that case, we don't need **server.passwd** file.

  remove password requirement

  ```
  cp server.key server.key.secure
  openssl rsa -in server.key.secure -out server.key
  ```

- Sign a certificate with a private key (self-signed certificate). Generate Certificate Signing Request(CSR).

  This will prompt a number of questions. However please ensure '**common name**' **matches** the host where your server is going to run.

  Generate Certificate Signing Request(CSR)

  ```
  openssl req -new -key server.key -out server.csr
  ```

- generate a self-signed certificate CRT, by using the CSR and private key.

Sign the certificate. server.crt must be accessible by client and server

```
openssl x509 -req -days 3650 -in server.csr -signkey server.key -out server.crt
```

- Generate dhparam file. ecFlow expects 2048 key.

```
openssl dhparam -out dh2048.pem 2048
```