SSL Certificates

NetYCE can be setup to use SSL certificates when accessed using the browser (http**s**). When https is enabled, the SSL protocol will also be used for the back-end communication between browser and the NetYCE API's and to access the REST API's (XCHapi and TMFapi).

Where the http protocol uses port 80, https will switch to port 443 forwarding any port 80 request to port 443.

> The back-end API port will use port 8080 for http and https. Although the 8080 port number can be changed, the 8080 port will only serve http or https depending on SSL the configuration.
>
> This http/https behaviour of port 8080 will soon be modified to support port 8080 for http and port 8443 for https.

The common procedure to install SSL certificates requires to first create a Certificate Signing Request (CSR) file which is then submitted to be signed by the Certificate Authority (CA). The resulting certificate (CRT) is installed on the server at the appropriate location. Once installed, the NetYCE configuration must then be updated to include the certificate and modify the used protocols.

To support this procedure, NetYCE includes cli tools to create the CSR and update the SSL configuration.

Create CSR

The **`mk_ssl_cert.pl`** tool is used to create the CSR in a few steps. First the SSL configuration needs to be defined, then the host KEY and CSR files can be generated. A third step allows for the creation of a PEM file that some signing procedures require.

The KEY file represents the PRIVATE KEY of the host. It is used for the SSL encryption and must be unique for the server. Normally the KEY file is generated once and should never be deleted or shared insecurely. The KEY file is also used to generate the CSR on which the CRT will be based. Losing or re-generating the KEY file of a server will render the CRT useless.

The `/opt/yce/system/mk_ssl_cert.pl` tool will prompt the user with a menu. Select option 1) to create or review the certificate configuration values.

```
$ mk_ssl_cert.pl
-----------------------------------------------------
Actions:
  1)  generate CONF file (certificate settings)
  2)  generate CSR file (certificate signing request)
  3)  extend   CSR into PEM and INFO files
  4)  generate CRT file (self-signed certificate)
  5)  extend   CRT into PEM and INFO files
  q)  quit
```

```
Select action:                                  [1]
```

The user is them prompted by some mandatory values. Most organizations will use some guidelines as to accepted values for these values. The responses will be stored in a ssl configuration file which will be used as defaults for later sessions: `/opt/yce/etc/ssl_cert.conf`

A sample session:

```
Select action:                                  [1]
  Please enter appropriate values for the certificate.
    (blank values will be ignored)
    C - Country Name (2 letter code)            [NL]
    ST - State or Province Name                 [Weesp] Hoord-Holland
    emailAddress - Email address                [yce@netyce.org]
    OU - Organizational Unit (eg section)       [development]
    CN - Common Name (fqdn)                     [genesis.netyce.org]
    O - Organization Name                       [NetYCE]
    L - Locality Name (eg city)                 [Weesp]
  -> created CONF file: /opt/yce/etc/ssl_cert.conf
-----------------------------------------------------
```

To create the CSR, select option 2). If a KEY file for the server name was found (in the directory /opt/yce/etc) a warning issued not to overwrite it.

```
Select action:                                  [1] 2
  Creating CSR
  -> have    KEY file: /opt/yce/etc/genesis.netyce.org.key
  WARNING: a KEY file already exists - overwriting will obsolete existing
certificates!
  Use existing KEY file?                        [Y]
```

If no KEY was found it will be created.

```
  Use existing KEY file?                        [Y] y
/usr/bin/openssl req -out /opt/yce/etc/genesis.netyce.org.csr -key
/opt/yce/etc/genesis.netyce.org.key -new -config /opt/yce/etc/ssl_cert.conf
  Set permissions
chmod 400 /opt/yce/etc/genesis.netyce.org.key
/opt/yce/etc/genesis.netyce.org.csr

  Below is the Certificate Signing Request (CSR) to submit to the
Certificate Authority (CA):
  Make sure the '-----BEGIN/END CERTIFICATE REQUEST-----' lines are
included.
  It can also be copied from '/opt/yce/etc/genesis.netyce.org.csr'.

-----BEGIN CERTIFICATE REQUEST-----
MIIDdDCCAlwCAQAwgZUxCzAJBgNVBAYTAk5MMSAwHgYDVQQDDBdkZXZlbDdhLmxl
ZnQubmV0eWNlLm9yZzEOMAwGA1UEBwwFV2Vlc3AxDzANBgNVBAoMBk5ldFlDRTEU
     ......
```

```
TX8swZ8/LkM6fbVeYOA6BMpIGT2RawRaoHjHydIc4vFLZx3W64ObmYA22RVRcwUm
FaAYE3znKD4qW+d76RikvjbKElNnZ+Mc1+yv5HVwUEH1lRIN2QV5h7FuM4FIJWh+
pr7D8JMviQvHOgcZOIRtunlLBzUIIlFA
-----END CERTIFICATE REQUEST-----


  -> used    KEY file: /opt/yce/etc/genesis.netyce.org.key
  -> created CSR file: /opt/yce/etc/genesis.netyce.org.csr
-------------------------------------------------------
```

Because signing by the CA requires the CSR and de KEY this is often combined in a PEM file. This file can be created using option 3).

```
Select action:                                  [3] 3
  Creating PEM
  Generating PEM from CSR
/usr/bin/openssl req -in /opt/yce/etc/genesis.netyce.org.csr -noout -text >
/opt/yce/etc/genesis.netyce.org.info
cat /opt/yce/etc/genesis.netyce.org.csr /opt/yce/etc/genesis.netyce.org.key
> /opt/yce/etc/genesis.netyce.org.pem
  Set permissions
chmod 400 /opt/yce/etc/genesis.netyce.org.info
/opt/yce/etc/genesis.netyce.org.pem
  -> created INF file: /opt/yce/etc/genesis.netyce.org.info
  -> created PEM file: /opt/yce/etc/genesis.netyce.org.pem
-------------------------------------------------------
```

Copy the content of the KEY, PEM and CSR files and submit as appropriate them for signing by the CA.

ssl_cert.conf

The generated CSR will include the now often required Subject Alternative Name (SAN) attributes. The `/opt/yce/etc/ssl_cert.conf` reflects this:

```
$ cat ssl_cert.conf
[req]
default_bits = 2048
distinguished_name = req_dn
prompt = no
req_extensions = req_ext

[req_dn]
C = NL
CN = genesis.netyce.org
L = Weesp
O = NetYCE
OU = development
ST = Weesp
emailAddress = yce@netyce.org
```

```
[req_ext]
basicConstraints = CA:FALSE
extendedKeyUsage = serverAuth, clientAuth
keyUsage = digitalSignature, nonRepudiation, keyEncipherment,
dataEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = genesis.netyce.org
```

It can be modified to include additional names/domains and ip-addresses like the example below:

```
[alt_names]
DNS.1 = genesis.netyce.org
DNS.2 = netyce.org
IP.1 = 172.17.0.24
IP.2 = 2003::24
```

Install CRT

Assuming the signed certificate was returned in CRT format (using `.crt` or `.cer` extensions), the file must be installed in the correct directory using the correct file names and activated.

If a PFX format was returned, see below for the extraction.

NetYCE expects the two SSL files in the directory `/opt/yce/etc`. They should be named after the full qualified name of the server using the extensions `.key` and `.crt` respectively.

As these files are highly private limited file permissions must be enforced.

```
cd /opt/yce/etc

sudo chown yce.nms genesis.netyce.org.key
sudo chown yce.nms genesis.netyce.org.crt

chmod 0400 genesis.netyce.org.key
chmod 0400 genesis.netyce.org.crt
```

PFX file extraction

Some signing authorities do not return a CRT (or CER) file but a PFX file. In that case the CRT file must be extracted from the FPX file.

```
openssl pkcs12 -in genesis.netyce.org.pfx -clcerts -nokeys -out
genesis.netyce.org.crt
```

The PFX file can also provide KEY file. But if this is the same that was used to create the CSR there is no need. If that was not the case (like a CRT for an entire domain), the KEY must be extracted in two steps and requires a decryption and encryption passphrase.

The CRT and KEY files obtained this way can be installed as described above

```
-- extract the key in encrypted form.
-- It first prompts for the decryption passphrase,
-- then twice for a new encryption passphrase
openssl pkcs12 -in netyce.org.pfx -nocerts -out netyce.org.encrypted.key

-- now extract the unencrypted KEY file
-- it will prompt for the new passphrase
openssl rsa -in netyce.org.encrypted.key -out genesis.netyce.org.key
```

Activating SSL certificate

The SSL certificate will be used by the web service (apache) and the back-end service (mojo). The NetYCE setup tool **`yce_setup.pl`** will be used to configure these services to enable SSL.

If the certificate only replaces an existing SSL CRT, the NetYCE application only needs to be restarted to have it activated. The yce_setup step can then be skipped.

To configure the local NetYCE server for SSL, start `yce_setup.pl` and continue to the “Yce server roles” section. Select the local server and answer each prompt. Answer 'yes' to the enable SSL prompt. Choose the SSL-hardening setting as per preference.

```
YCE servers currently in setup:
1) genesis.netyce.org (*)
   | IPv4-address          | IPv6-address
   |  172.17.0.24          |  3001::24
  local server is marked with (*)
  Select the server-number to Edit/Remove, or 'A' to add, 'C' to continue:
[C]

YCE server roles:
1) genesis.netyce.org (*)
   | Front-end     | SSL      | URL       | Backend
   |  yes          |  http    |  name     |  8080
   | Database      | Db-id
   |  yes          |  1
  local server is marked with (*)
  Select the server-number to change, 'C' to continue: [1]
    'genesis' is (also) a Front-end server?   [yes]
    'genesis' is DNS resolvable (y/n)?        [yes]
    'genesis' uses SSL (y/n)?                 [no] ?

  The use of SSL or 'secure-socket-layer' is highly recommended for
```

```
production systems.
  It requires the generation and signing of a server-certificate by the CA
(certificate-
  authority) of your company.
  More information on the generation of a SSL certificate on NetYCE servers
is located at:
'https://wiki.netyce.com/doku.php/maintenance:general:tools:mk_ssl_cert.pl'

    'genesis' uses SSL (y/n)?                [no] y
    'genesis' uses SSL-hardening (y/n)?      [no] ?

  SSL can be setup to accept older (weaker) levels of TLS (transport-layer-
security)
  as well as the newer (hardened) level of TLS1.2. When selecting 'SLL-
hardening'
  the http server will only accept connections supporting TLS1.2 and reject
older
  levels.

    'genesis' uses SSL-hardening (y/n)?      [no] y
    'genesis' portnumber of backend server?  [8080]
    'genesis' is (also) a Database server?   [yes]
    'genesis' uses database-id value (1/2)?  [1]
```

Service restart

use the commands `go restart httpd -f` and `go restart mojo -f` to restart these services. The `-f` option is used to force the restart instead of the standard hot-deploy.

```
$ go restart httpd -f
-- restarting Daemon 'httpd'
httpd: 1022 11920 11923 11924 11925
  stop: /usr/bin/sudo /usr/bin/systemctl stop httpd.service
  wait stop 'httpd':
  start: /usr/bin/sudo /usr/bin/systemctl start httpd.service
  wait start 'httpd': 12272 12273 12274 12275 12276
done

$ go restart mojo -f
-- restarting Daemon 'mojo'
mojo: 11990 12058 12059 12060 12061 12062 12063
  stop: /opt/yce/system/init/yce_mojo stop
  wait stop 'mojo':
  start: /opt/yce/system/init/yce_mojo start
  wait start 'mojo': 12161 12162 12163 12164 12165 12166 12167
done
```