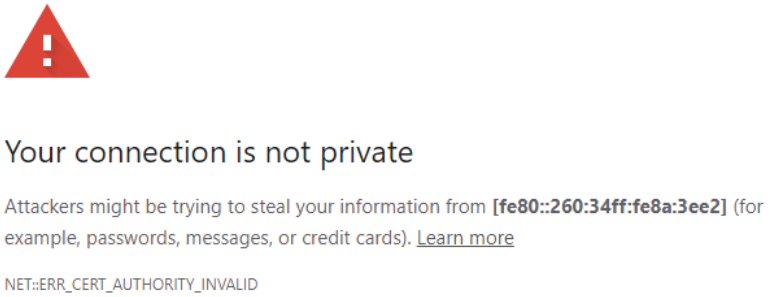
QUICK START GUIDE
Generating SAN Certificates for ctrlX CORE

GETTING STARTED

This guide describes a simple procedure for generating self-signed SAN certificates for ctrlX CORE. In our case, the Secondary Alternate Name (SAN) will specify the IP address of the control. Such certificates will circumvent the error (ERR_CERT_AUTHORITY_INVALID) generated by the default certificate:



1 openssl

openssl.exe from the OpenSSL Project[1] will be used to generate the certificates. This application is distributed as standard in many Linux distributions. It is also included with Git[2]. On Windows, adding *openssl.exe* via Git may be the simplest option. Otherwise it is possible to build it for Windows from source.

The instructions below have been tested on Ubuntu 18.04 (virtual machine) and Ubuntu 20.04 (native OS). They have also been tested in a Git Bash shell on Windows.

2 Private key

In a terminal window enter the following command:

```
openssl genrsa -out webserver_custom_key.pem 2048
```

This will create a file in the local folder called *webserver_custom_key.pem* with the required private key. Do not change the filename.

[1] https://www.openssl.org/
[2] https://git-scm.com/downloads

3 Certificate signing request (CSR)

Next we create a certificate signing request based on our private key with the following command:

```
openssl req -new -key webserver_custom_key.pem -out webserver_custom.csr
```

Here *webserver_custom.csr* is a temporary file that will be used in subsequent steps. As the command runs you will be prompted for certain information. Enter appropriate data for all fields, leaving the common name field *.

```
Country: US
State: IL
Locality: HES
Company: Bosch Rexroth
Organization: DCNA/SAE22-US
Common name: *
Email: admin@bosch.com
```

4 Additional data

We also require an additional temporary file, here called *v3.ext*, with the following content:

```
subjectKeyIdentifier   = hash
authorityKeyIdentifier = keyid:always,issuer:always
basicConstraints       = CA:TRUE
keyUsage               = digitalSignature, keyCertSign
subjectAltName         = IP:192.168.1.1, IP:192.168.100.101
issuerAltName          = issuer:copy
```

Edit the IP address in line *subjectAltName* to match the IP address of your control. You may enter multiple addresses separated by commas as shown. Use any available text editor (e.g. Notepad, Nano) to create the file.

4 Creating the certificate

Finally, create the certificate, *webserver_custom_cert.pem*, with the command below (enter as a single line):

```
openssl x509 -req -in webserver_custom.csr -signkey webserver_custom_key.pem
  -out webserver_custom_cert.pem -days 3650 -sha256 -extfile v3.ext
```

5 ctrlX CORE

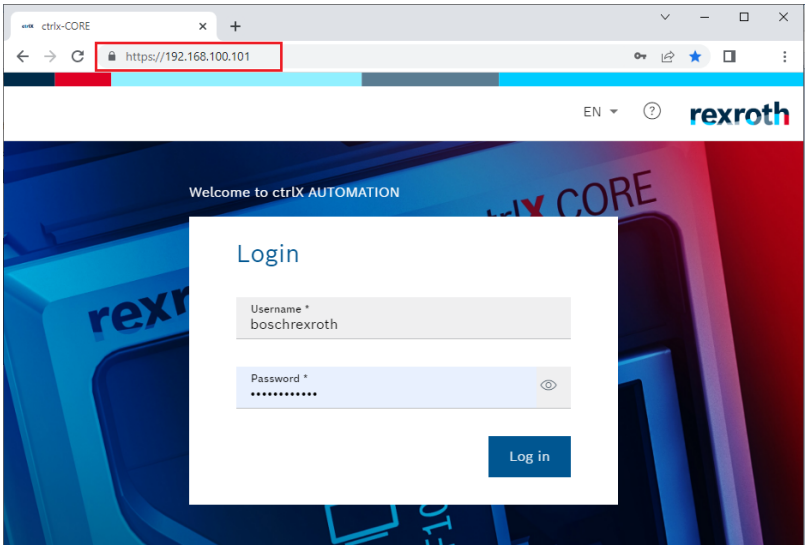
Using the web UI of the ctrlX CORE, navigate to the certificate manager (see Settings>Certificates & Keys>Web Server) and replace files *webserver_custom_cert.pem* and *webserver_custom_key.pem* with the similarly named files created in the previous steps. When importing the files, you will be prompted to define the certificate's category. Select category *Own*.

Reboot the control.

6 Chrome

In Chrome[3], open the security settings (chrome://settings/security) and select the certificate manager. Import the custom certificate (*webserver_custom_cert.pem*) as type *Trusted root authority*.

If the configuration is successful, Chrome will no longer indicate a problem with the certificate when connecting to the control's web UI.



[3] Version tested: 101.0.4951.67 (Official Build) (64-bit)