NETWORK TECHNOLOGIES INCORPORATED
1275 Danner Dr Tel:330-562-7070
Aurora, OH 44202 Fax:330-562-1999
www.networktechinc.com

Creating an x.509 CA certificate for ENVIROMUX Series Products

The ENVIROMUX family of products are designed to be configurable with security to limit access to their web interface controls. The use of x.509 client authentication is one of the methods that may be used, and although the ENVIROMUX includes a default x.509 CA certificate, this procedure will help you create your own custom x.509 CA certificate to use with this feature. This procedure was created using Ubuntu Linux and OpenSSL.

Note: Do not disable access to the ENVIROMUX web interface using http before you verify that the https client authentication works properly (see last page).

Creating a Certificate Authority using OpenSSL

The Root CA certificate will be used by a web server (ENVIROMUX) to authenticate the client (browser). It also needs to be imported in a web browser as a Trusting authority.

An example SSL config file is attached as **`openssl.cnf .`** **(You can edit it in any text editor to customize for your own needs.)**

Creating the Certificate Management Directories and Files

1. Create directory “ntiCA” in /usr/local/ssl for ntiCA certificate management and change to that directory.

(“nti” can be changed to whatever you want throughout this procedure, but do it consistently. Whatever you change it to, make sure the openssl.cnf file is edited to match your changes)

```
mkdir /usr/local/ssl/ntiCA
cd /usr/local/ssl/ntiCA
```

Create following directories in the ntiCA directory:

```
mkdir CA
mkdir server
mkdir server/certificates
mkdir server/requests
mkdir server/keys
mkdir user
mkdir user/certificates
mkdir user/requests
mkdir user/keys
```

The CA directory will be populated with the certificate authority certificate request, keys and certificate used to sign server and user certificates. The server directory hierarchy will be used to manage certificate requests, keys and certificates issued for web server hosts. The user directory hierarchy will be used to manage certificate requests, keys and certificates for users.

2. Issue the following commands to setup default contents of certificates and revocation list for these files:

(The percent sign (%) is the command prompt, not part of the command.)

```
% cd /usr/local/ssl/ntiCA
% echo "01" > serial
% touch index.txt
```

The openssl.cnf file that you edited earlier (if you did) references these files so make sure they are created in the ntiCA directory.

Creating the ntiCA Key and Certificate

The general process for creating a certificate includes:

1. Creating a private key
2. Creating a certificate request
3. Creating and signing a certificate from the certificate request

1. Create the CA key:

```
% cd /usr/local/ssl/ntiCA
% openssl genrsa -out ./CA/ntiCA.key 2048
Generating RSA private key, 2048 bit long modulus
.....++++++
......++++++
e is 65537 (0x10001)
```

2. Create the CA certificate request:

```
% openssl req -sha512 -new -key ./CA/ntiCA.key -out ./CA/ntiCA.csr
```

You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a **Distinguished Name** or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

```
Country Name (2 letter code) [US]:
State or Province Name (full name) [OH]:
Locality Name (eg, city) [Aurora]:
Organization Name (eg, company) [NTI]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:your_user_name
Email Address [sales@ntigo.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:
```

3. Self-sign the CA certificate:

```
% openssl x509 -req -sha512 -days 3650 -in ./CA/ntiCA.csr -out ./CA/ntiCA.crt -signkey
./CA/ntiCA.key
Signature ok
Getting Private key
```

Verifying the CA certificate contents

At this point we have our self-signed CA certificate and our CA key, which will be used to sign the web server and client certificates that we create. To verify the certificate contents, use the following command:

```
% openssl x509 -in ./CA/ntiCA.crt -text
```

Creating a Web Server Certificate (This will need to be done for each web server)

The procedure for creating a web server certificate is similar to that for creating the CA certificate except that the web server certificate will be signed using the CA key rather than self-signing with a web server-specific key.

1. Create the web server private key using a fully qualified DNS name (or IP address). When prompted for the pass phrase, **enter a password that you can remember.**

```
% cd /usr/local/ssl/ntiCA
% openssl genrsa -des3 -out ./server/keys/your_device_fqdn_or_ipaddress.key 2048
Generating RSA private key, 2048 bit long modulus
.......++++++
.++++++
e is 65537 (0x10001)
Enter pass phrase for ./server/keys/your_device_fqdn_or_ipaddress.key:
Verifying - Enter pass phrase for ./server/keys/your_device_fqdn_or_ipaddress.key:
```

2. Create the web server certificate request using the same fully qualified DNS name (or IP address) you used for the private key. When prompted for the pass phrase for the keys in file ./server/keys/your_device_fqdn_or_ipaddress.key, enter the pass phrase that you used for the private key. Also, **it is vitally important** that you set the Common Name value to the fully qualified DNS name of your web server because that's the value that a browser client will verify when it receives the web server's certificate.

```
% openssl req -sha512 -new -key ./server/keys/your_device_fqdn_or_ipaddress.key -out
./server/requests/your_device_fqdn_or_ipaddress.csr
Enter pass phrase for ./server/keys/your_device_fqdn_or_ipaddress.key:
```

You are about to be asked to enter information that will be incorporated into your certificate request.
What you are about to enter is what is called a **Distinguished Name** or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

```
Country Name (2 letter code) [US]:
State or Province Name (full name) [OH]:
Locality Name (eg, city) [Aurora]:
Organization Name (eg, company) [NTI]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:your_device_fqdn_or_ipaddress
Email Address [ca@ntigo.com]:sales@ntigo.com

Please enter the following 'extra' attributes
  to be sent with your certificate request
  A challenge password []:
  An optional company name []:
```

3. Sign the web server certificate with the CA key:

```
% openssl ca -days 3650 -in server/requests/your_device_fqdn_or_ipaddress.csr -cert
./CA/ntiCA.crt -keyfile ./CA/ntiCA.key -out
./server/certificates/your_device_fqdn_or_ipaddress.crt -config <path_to_config
file>\openssl.cnf
```

In the command above, substitute the path to the config file “openssl.cnf” in place of “<path_to_config_file>”.

```
DEBUG[load_index]: unique_subject = "yes"
  Check that the request matches the signature
  Signature OK
  Certificate Details:
  Serial Number: 3 (0x3)
  Validity
  Not Before: Aug 18 17:41:07 2005 GMT
  Not After : Aug 18 17:41:07 2006 GMT
  Subject:
  countryName = US
  stateOrProvinceName = OH
  organizationName = NTI
  commonName = your_device_fqdn_or_ipaddress
  emailAddress = sales@ntigo.com
  X509v3 extensions:
  X509v3 Basic Constraints:
  CA:FALSE
  Netscape Comment:
  OpenSSL Generated Certificate
  X509v3 Subject Key Identifier:
  0A:6B:79:E7:98:5F:30:7F:A0:67:4A:12:83:9C:0A:58:BE:8B:41:2A
  X509v3 Authority Key Identifier:
  DirName:/C=US/ST=OH/L=Aurora/O=NTI /CN=NTI CA/emailAddress=sales@ntigo.com
  serial:CD:93:0B:9F:5A:71:EB:8B

  Certificate is to be certified until Aug 18 17:41:07 2026 GMT (365 days)
  Sign the certificate? [y/n]:y

  1 out of 1 certificate requests certified, commit? [y/n]y
  Write out database with 1 new entries
  Data Base Updated
```

To verify the web server certificate contents, use the following command:

```
% openssl x509 -in ./server/certificates/your_device_fqdn_or_ipaddress.crt -text
```

Key values to look for are:

```
Subject CN=your_device_fqdn_or_ipaddress
Issuer CN=NTI CA
```

Uploading Server Certificate to NTI device

The NTI ENVIROMUX webserver expects the certificate and key as a single file in "PEM" format.

Note: If your key has a password then you need to create a key without password.

Use the following command to export the file without the password.

```
openssl rsa -in <your_key>.key -text > private.key
```

Use following command to create pem certificate file

```
cat <your_certficate_name>.crt private.key > <server_name>.pem
```

On the ENVIROMUX WEB Interface menu Under “Administration” select “Security”.

In X509 certificates

Select the above file and press the button **"Upload Server certificate and Key"**

<your_key> , <your_certificate_name> and <server_name> are placeholders. “Your_ certificate” is the web server certificate you created, “your_key” is the CA key you created, and the “server_ name” is whatever you want the pem file to be named.

Creating a Client Certificate

The procedure for creating a client certificate is similar to that for creating the web server certificate.

Creating a user key

The following instructions create a private key for a user named your_name@ntigo.com. When prompted for the pass phrase, enter a password that you can remember.

```
% cd /usr/local/ssl/ntiCA
% openssl genrsa -des3 -out ./user/keys/your_name@ntigo.com.key 2048
Generating RSA private key, 2038 bit long modulus
...++++++
.....++++++
e is 65537 (0x10001)
Enter pass phrase for ./user/keys/your_name@ntigo.com.key:
Verifying - Enter pass phrase for ./user/keys/your_name@ntigo.com.key:
```

Create the user certificate request

1. The following command creates a certificate request for a user with email address: your_name@ntigo.com and common name your_name. When prompted for the pass phrase for the keys in file ./user/keys/your_name@ntigo.com.key, enter the pass phrase that you used to create the user key (e.g. “password”).

```
% openssl req -sha512 -new -key ./user/keys/your_name@ntigo.com.key -out
./user/requests/your_name@ntigo.com.csr
Enter pass phrase for ./user/keys/your_name@ntigo.com.key:
```

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a **Distinguished Name** or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

```
Country Name (2 letter code) [US]:
  State or Province Name (full name) [OH]:
```

```
Locality Name (eg, city) [Aurora]:
Organization Name (eg, company) [NTI]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:your_name
Email Address [ca@ntigo.com]:your_name@ntigo.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

2. Sign the user certificate request and create the certificate

```
% openssl ca -in ./user/requests/your_name@ntigo.com.csr -cert ./CA/ntiCA.crt -keyfile
./CA/ntiCA.key -out ./user/certificates/your_name@ntigo.com.crt
```

Using configuration from /usr/local/ssl/openssl.cnf

```
DEBUG[load_index]: unique_subject = "yes"
```

3. Check that the request matches the signature

```
Signature OK
Certificate Details:
Serial Number: 4 (0x4)
Validity
Not Before: -----
Not After : -----
Subject:
countryName = US
stateOrProvinceName = OH
organizationName = NTI
commonName = your_name
emailAddress = your_name@ntigo.com
X509v3 extensions:
X509v3 Basic Constraints:
CA:FALSE
Netscape Comment:
OpenSSL Generated Certificate
X509v3 Subject Key Identifier:
 -----
X509v3 Authority Key Identifier:
DirName:/C=US/ST=OH/L=Aurora/O=NTI/CN=your_nameCA/emailAddress=sales@ntigo.com
serial:CD:93:0B:9F:5A:71:EB:8B
----
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

Verifying the user certificate contents

To verify the user certificate contents, you can use the following command:

```
% openssl x509 -in ./user/certificates/your_name@ntigo.com.crt -text
```

Importing a Client Certificate into Web Browsers

Web browsers like Firefox and IE can't use the certificates in the PEM format that is generated by OpenSSL . Consequently, we'll need to export the user certificate to file formats that can be imported by web browsers.

Importing the client certificate in PKCS#12 format

Firefox and Internet Explorer 6.0 support the PKCS#12 certificate format. Use the following command to convert the user certificate to this format.

NOTE: During the conversion process, you'll be asked for an export password. Enter anything you can remember, but don't let it be empty because the file will contain your private key.

```
    % openssl pkcs12 -export -clcerts -in ./user/certificates/your_name@ntigo.com.crt -inkey
./user/keys/your_name@ntigo.com.key -out ./user/certificates/your_name@ntigo.com.p12
```

Copy the `your_name@ntigo.com.p12` file to a location where you can access it from your web browser via the file system.

Import Using Internet Explorer 6.0

To import a certificate, start IE and follow the instructions below:

Navigate to the Tools menu and click Internet Options
Click the Content tab
Click the Certificates button
Click the Import button
Follow the wizard instructions to select the certificate file
Enter the password you used to protect your certificate and private key
Import client certificates into the Personal store and root certificates for the CA that signed the web server certificates into the Trusted Root Certification Authorities store
Click the imported certificate and then on the View button in the Certificate intended purposes group box. Click the Details tab and then the Edit Properties button. Make sure that the Client Authentication option is checked.

For more detailed information, please see Microsoft Internet Explorer 6 Resource Kit, Chapter 6 - Digital Certificates.

Import using FireFox 1.5

To import a certificate, start FireFox and follow the instructions below:

Navigate to the Tools menu and click Options
Click the Advanced icon
Click the Security tab
Click the View Certificates button
Click the Import button and select the certificate file
Enter your master password for the Software Security Device
Enter the password you used to protect your certificate and private key

Importing the nti CA root certificate into web browsers

In order to establish a chain of trust between the imported user certificate and the issuing certificate authority, you'll need to import the nti CA certificate into your web browser.

Though the user interface for accepting the CA certificate varies, it is possible to import it for Firefox and IE 6.0 in this way.

Firefox 1.5

A dialog box appears and offers the choice of importing the CA certificate. Select the “Trust this CA” to identity web sites option, then click the “OK” button. You may also select the “View” button to see the certificate contents before accepting it.

Internet Explorer 6.0

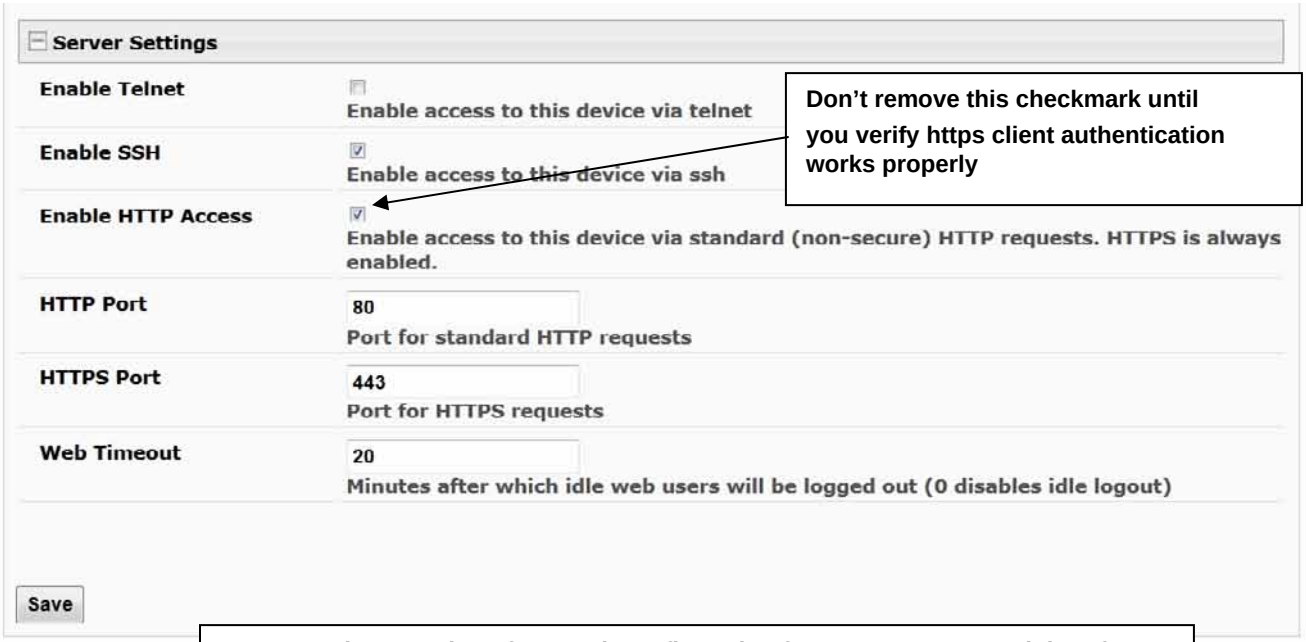
A dialog box appears and asks “Do you want to open or save this file?”. Select the “Open” option, then click the “Install Certificate” button when the certificate dialog appears.

Once you've successfully imported the nti CA you will be able to access the URL of the ENVIROMUX without being prompted to accept the web server certificate.

Configuring NTI device to require Client Certificate

On the ENVIROMUX WEB Interface menu Under “Administration” select “Security”.
In X509 certificates select the file `ntiCA.crt` and press button "Upload CA certificate"
To enable the device to ask for client certificate select “certificate + login” in the "Mode" field under “User Authentication”.
Use https communication.

Note: Before disabling http be sure to verify https client authentication works properly.



Server settings section of Network configuration from ENVIROMUX web interface