How to perform peer verification with clients connecting to LS Message Broker (RabbitMQ)

Last Modified on 09/26/2024 5:56 pm EDT

Procedure Steps

Under normal circumstances, a Certificate Authority (CA) would generate the server certificates and client certificates. Follow these steps to create a CA, server, and client certificates using OpenSSL:

Install and configure OpenSSL.

a. Generate OnGuard Root CA. In the following example, a folder has been created in the C: drive called "certs". Run the following command and supply appropriate values:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout C:\certs\CA.key –out C:\certs\CA.crt
```

b. Convert the CA certificate to PEM format:

```
openssl x509 -in C:\certs\CA.crt -out C:\certs\CA.pem -outform PEM
```

c. Generate a Generic Common Server Certificate Request. Issue it to the server OU name than the CA.
Note: To generate new client and server certificates, start from this step with the existing CA.

```
openssl req -out C:\certs\server.csr -newkey rsa:2048 -nodes -keyout C:\certs\server.key
```

d. Generate the Common Server Certificate. The req section of the OpenSSL configuration file must have the following values in the request section, appropriate values can be supplied for state, locality, and so on. In this example, a separate file was created called openssl_client.cfg with only the values below in the configuration file, and the file is specified in the command to generate the server certificate. There is also an optional section which includes subject alternative names for the certificate and / or IP addresses:

```
[req]
distinguished_name = req_distinguished_name
req_extensions  = v3_req
x509_extensions = usr_cert
[req_distinguished_name]
countryName = Country Name (2 letter code)
countryName_default = US
stateOrProvinceName = State or Province Name (full name)
stateOrProvinceName_default = NY
localityName = Locality Name (eg, city)
localityName_default = Locality
organizationalUnitName = Organizational Unit Name (eg, section)
organizationalUnitName_default = OU-Name
commonName = ServerName
commonName_max = 64
[ v3_req ]
# Extensions to add to a certificate request
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
```

```
extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection
subjectAltName = @alt_names
extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection

[alt_names]
DNS.1   = www.example.com
DNS.2   = *.domain.com
IP.1  = 10.10.10.10

[ usr_cert ]
basicConstraints=CA:FALSE
nsCertType = client, server, email
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection
nsComment = "Sample CA Signed Cert"
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer
```

```
openssl x509 -req -days 500 -sha256 -in C:\certs\server.csr -CA C:\certs\CA.crt -CAkey C:\certs\CA.key -CAcreateserial -out C:\certs\server.pem -extensions v3_req -extfile "C:\Program Files\OpenSSL-Win64\bin\openssl_client.cfg"
```

e. Convert the Server Certificate to .pfx format. Leave the export password blank.

```
openssl pkcs12 -inkey C:\certs\server.key -in C:\certs\server.pem -export -out C:\certs\client.pfx
```

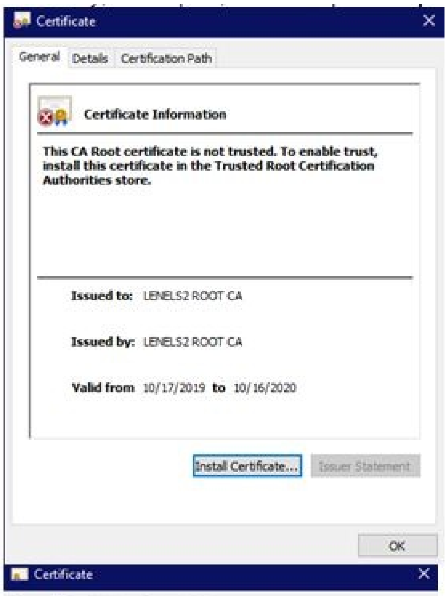
f. Convert client.pfx to a PEM format so that it can be converted to the .p12 format required by OnGuard clients and services:

```
openssl pkcs12 -in C:\certs\client.pfx -nodes -out C:\certs\temp.pem
```

g. Convert temp.pem to .p12 format. Leave the export password blank:

```
openssl pkcs12 -export -in C:\certs\temp.pem  -out C:\certs\client.p12
```

1. Ensure the CA is installed and trusted by the host machine(s) that will be using them in the Trusted Root Certification Authorities Store. In this example, copy CA.crt, server.key, server.pem, and client.p12 to %programdata%\lnl\nginx\conf:



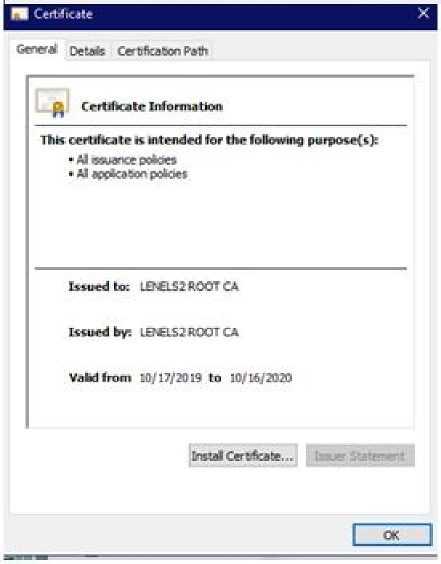
Certificate Information

This CA Root certificate is not trusted. To enable trust, install this certificate in the Trusted Root Certification Authorities store.

Issued to: LENELS2 ROOT CA

Issued by: LENELS2 ROOT CA

Valid from 10/17/2019 to 10/16/2020



Certificate Information

This certificate is intended for the following purpose(s):

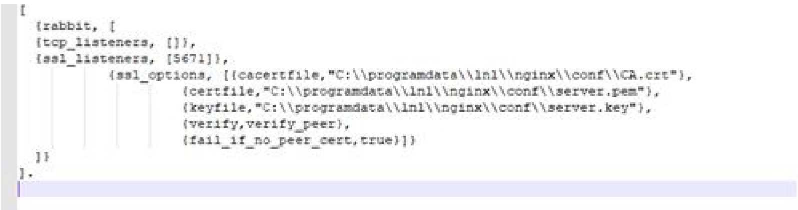
- All issuance policies
- All application policies

Issued to: LENELS2 ROOT CA

Issued by: LENELS2 ROOT CA

Valid from 10/17/2019 to 10/16/2020

2. Stop LS Message Broker, RabbitMQ, LS Event Context Provider, LS Communication Server, LS Open Access, LS Web Event Bridge, and LS Web Service services.
3. In Task Manager, ensure that the epmd.exe process and erl.exe processes are stopped.
4. Open %programdata%\lnl\\RabbitMQ\advanced.config file in a text editor.
5. Update fail_if_no_peer_cert, false to true, update cacertfile path, certfile path, and keyfile path. In this example, the files were copied to the default certificate location in %programdata%\lnl\nginx.conf:



```
[
  {rabbit, [
  {tcp_listeners, []},
  {ssl_listeners, [5671]},
            {ssl_options, [{cacertfile,"C:\\programdata\\lnl\\nginx\\conf\\CA.crt"},
                        {certfile,"C:\\programdata\\lnl\\nginx\\conf\\server.pem"},
                        {keyfile,"C:\\programdata\\lnl\\nginx\\conf\\server.key"},
                        {verify,verify_peer},
                        {fail_if_no_peer_cert,true}]}
  ]}
].
```

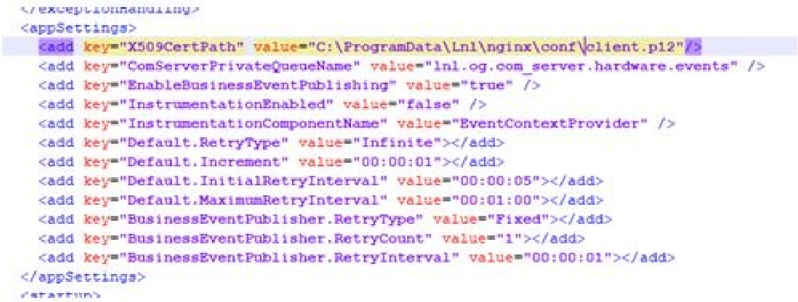
6. Add the following to the acs.ini file on every client and server machine along with the client.p12 certificate, where CertificatePath equals the path to the .p12 certificate created in step 1:

`[LSMessageBrokerClientSecurity]`

`CertificatePath="C:\ProgramData\Lnl\nginx\conf\client.p12"`

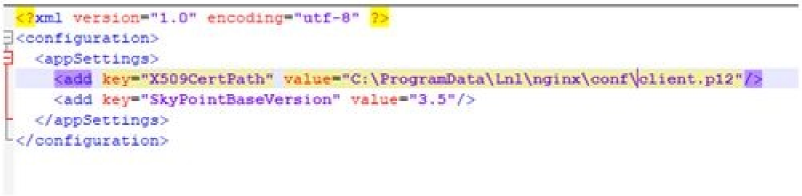
7. Modify xml configuration files for Lnl.OG.EventContextProviderService.exe.config, LnlComsrvr.exe.config, Lnl.OG.LsOpenAccess.exe.config, and Lnl.OG.WebEventBridgeService.exe.config to include the X509 certificate path key location.
8. Add the following line to the configuration files:
   Note: Failure to correctly modify these files can cause these services to not start.

Event Context Provider:



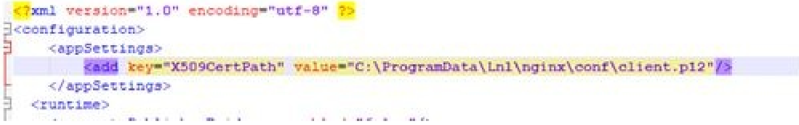
```
</exceptionHandling>
<appSettings>
  <add key="X509CertPath" value="C:\ProgramData\Lnl\nginx\conf\client.p12"/>
  <add key="ComServerPrivateQueueName" value="lnl.og.com_server.hardware.events" />
  <add key="EnableBusinessEventPublishing" value="true" />
  <add key="InstrumentationEnabled" value="false" />
  <add key="InstrumentationComponentName" value="EventContextProvider" />
  <add key="Default.RetryType" value="Infinite"></add>
  <add key="Default.Increment" value="00:00:01"></add>
  <add key="Default.InitialRetryInterval" value="00:00:05"></add>
  <add key="Default.MaximumRetryInterval" value="00:01:00"></add>
  <add key="BusinessEventPublisher.RetryType" value="Fixed"></add>
  <add key="BusinessEventPublisher.RetryCount" value="1"></add>
  <add key="BusinessEventPublisher.RetryInterval" value="00:00:01"></add>
</appSettings>
<startup>
```

Lnlcomsrvr.exe:



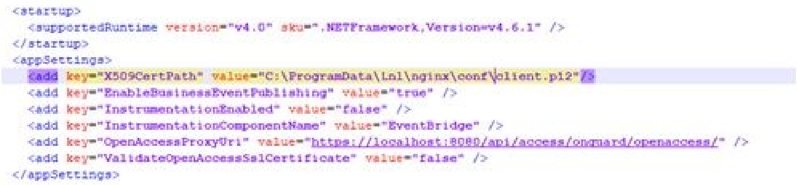
```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="X509CertPath" value="C:\ProgramData\Lnl\nginx\conf\client.p12"/>
    <add key="SkyPointBaseVersion" value="3.5"/>
  </appSettings>
</configuration>
```

Lnl.OG.LsOpenAccess.exe.config:



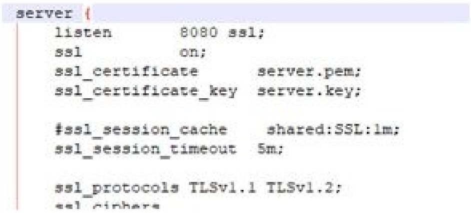
```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <appSettings>
        <add key="X509CertPath" value="C:\ProgramData\Lnl\nginx\conf\client.p12"/>
    </appSettings>
  <runtime>
```

Lnl.OG.WebEventBridgeService.exe.config:



```
<startup>
  <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6.1" />
</startup>
<appSettings>
  <add key="X509CertPath" value="C:\ProgramData\Lnl\nginx\conf\client.p12"/>
  <add key="EnableBusinessEventPublishing" value="true" />
  <add key="InstrumentationEnabled" value="false" />
  <add key="InstrumentationComponentName" value="EventBridge" />
  <add key="OpenAccessProxyUri" value="https://localhost:8080/api/access/onguard/openaccess/" />
  <add key="ValidateOpenAccessSslCertificate" value="false" />
</appSettings>
```

Modify the nginx.conf file to include the new cert names/location located in %programdata%\lnl\nginx\conf\nginx.conf:



```
server {
    listen       8080 ssl;
    ssl          on;
    ssl_certificate      server.pem;
    ssl_certificate_key  server.key;

    #ssl_session_cache    shared:SSL:1m;
    ssl_session_timeout  5m;

    ssl_protocols TLSv1.1 TLSv1.2;
    ssl_ciphers
```

Ensure that the Message Broker Host and Open Access Host settings in **System Administration >**

System Options matches the certificate FQDN.

Restart LS Message Broker, LS Communication Server, LS Event Context Provider, LS Open Access, and LS Web Event Bridge.

Applies To

OnGuard 7.5 or later

Additional Information

Supporting RMQ documentation:

https://www.rabbitmq.com/ssl.html

https://www.rabbitmq.com/ssl.html#peer-verification