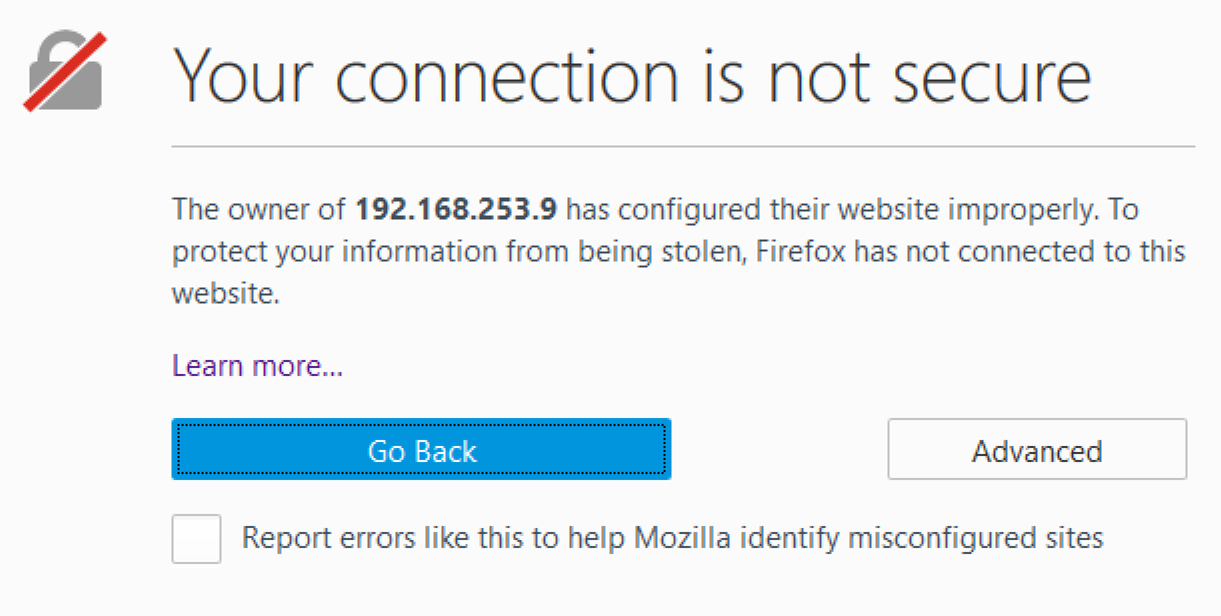
Server Authentication

Lumeta comes with a self-signed certificate that allows Secure Sockets Layer (SSL) for the web GUI out of the box. Best practice is to request and install a certificate from a trusted Certificate Authority (CA) that verifies the authenticity of the Lumeta system and avoids users from receiving warning messages like the one below:



Your connection is not secure

The owner of **192.168.253.9** has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

Learn more...

□ Report errors like this to help Mozilla identify misconfigured sites

Server Authentication

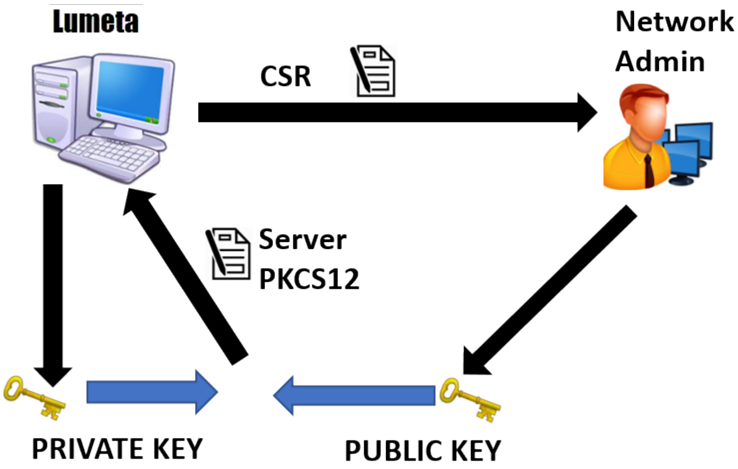
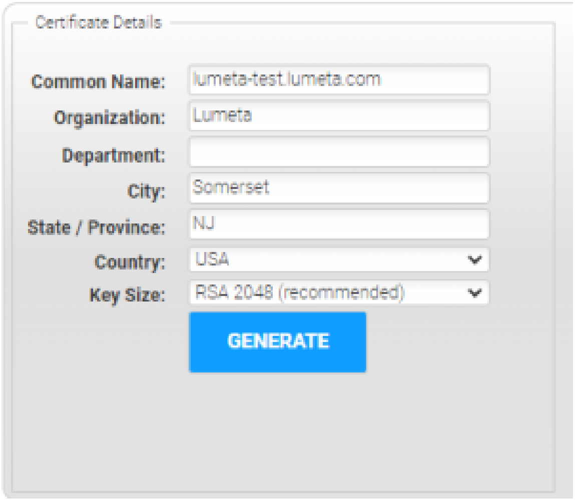


Figure 2: Sever Authorization

Generating a Certificate Signing Request

The first step in obtaining a server certificate is creating a Certificate Signing Request (CSR) file. Speak with your Network Admin on the preferred method to create the CSR.

Lumeta supports openssl. You can use an online tool like Digicert OpenSSL CSR Generator to obtain the openssl line command needed to create the CSR file.



Certificate Details

Common Name:	lumeta-test.lumeta.com
Organization:	Lumeta
Department:	
City:	Somerset
State / Province:	NJ
Country:	USA
Key Size:	RSA 2048 (recommended)

Will generate this command: o*penssl req -new -newkey rsa:2048 -nodes -out lumeta-test_lumeta_com.csr -keyout lumeta-test_lumeta_com.key -subj " /C=US/ST=NJ/L=Somerset/O=Lumeta/CN=lumeta-test.lumeta.com"*

Log into your Lumeta system. Type support bash at the CLI to enter the shell prompt. Copy and paste your unique openssl command and hit enter. Two files will be generated:

-the csr file.

-your private key.

Make note of the directory with the pwd command. Send the csr file to your network admin and request the CA certificate be in pem format.

Convert Server Certificate Files into p12 bundle

Lumeta requires a certificate in the form of a p12 bundle of the CA server certificate, private key, and intermediate key (if applicable). The p12 bundle must have a friendly name and password,

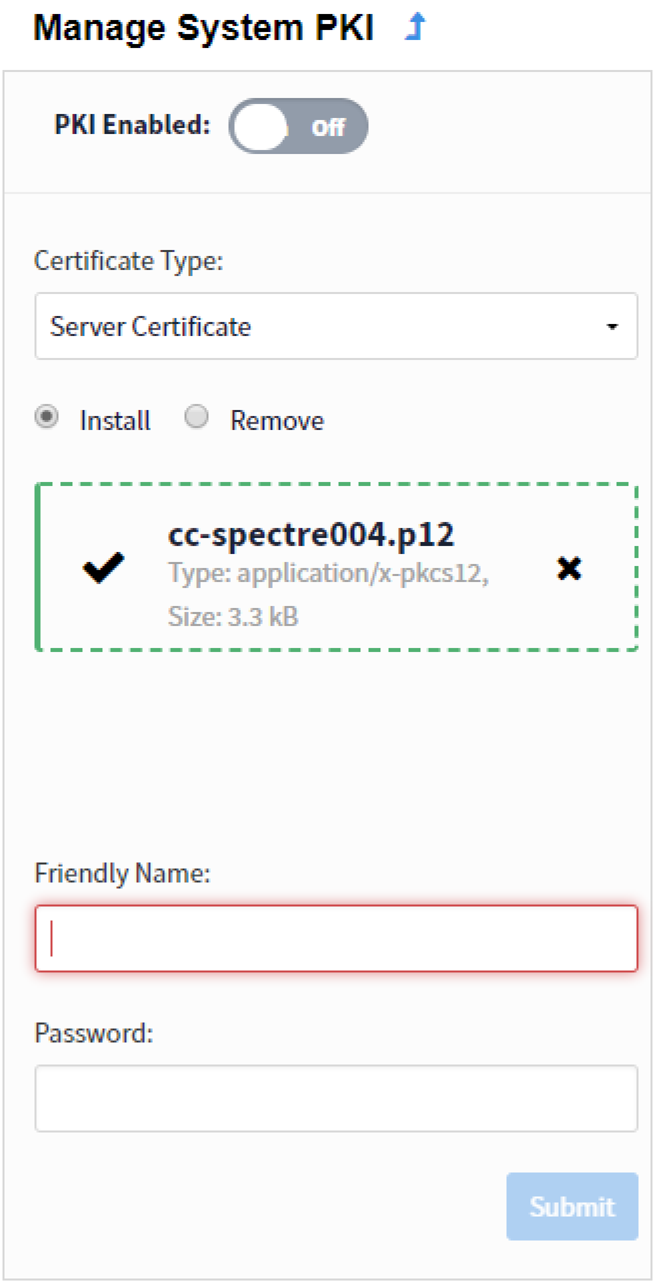
Please discuss with your Network Admin if you can get the certificate in this format. If unable, then request the certificates in pem format and follow these steps.

Copy the two file to the Lumeta System in same directory as the private key generated in above step.

1. Bundle the Command Center Private Key and the newly formatted Command Center Public Key into a p12 file.
   a. Login to the Lumeta System and type support bash.
   b. cd to the directory of the private key, server certificate, and Intermediate certificate(if applicable).
   c. Type the below command updating with the appropriate file names. Leave out single quotes and for friendly name supply your unique word.
      openssl pkcs12 -export -in 'server-certificate' -inkey 'private-key' -certfile 'intermediate-certificate' -out 'lumeta-server-certificate.p12' -name 'friendly-name'
   d. Remember the friendly name you defined. You will need this later.
   e. User will be prompted to enter the pass phrase for the private.key. Also the user will need to provide twice the Export Password.
   f. A new certificate in p12 format has now been created. This is your new server certificate.

Installing the Server Certificate

1. Through CLI: On the Command Center CLI type the following command to install the certificate:
   certificate server install "pathto/file/filename" "friendly-name" "private.key password"
2. Through WEB UI:
   a. Copy the "CC-ipaddress-pkcs12".p12 off the Command Center to your directory.
   b. On the UI navigate to Lumeta Systems and Manage PKI.
   c. Select Server Certificate from the Certificate Type. Upload the Certificate and input the Friendly Name and Password.



Manage System PKI

PKI Enabled: Off

Certificate Type:

Server Certificate

● Install ○ Remove

✓ cc-spectre004.p12
Type: application/x-pkcs12,
Size: 3.3 kB ✖

Friendly Name:

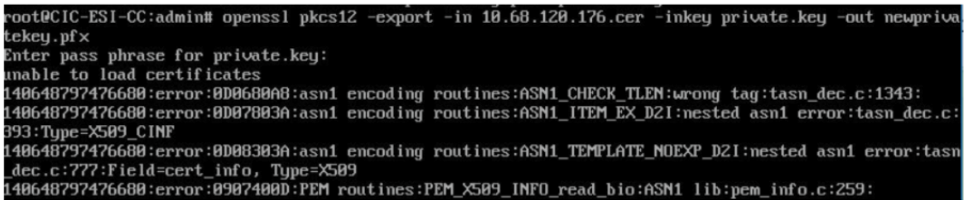
Password:

APPENDIX A: Verifying Certificates

1. Verify the subject line of the CA-public.pem file matches the issuer line of the public-user.cer file using these openssl commands.
   openssl x509 -in public-user.cer -noout -subject -issuer
   openssl x509 -in CA-chain.pem -noout -subject -issuer
2. You can change the extension of the "public-user.cer" file to "public-user.txt" to view the certificate in notepad. Then this public-user certificate can be verified by comparing it to the "public-user".cer in the database by running this db command.
   select * from system.user_certificate;
3. The CA certificate can be verified in /etc/pki/lumeta folder. There will be a file 'httpd_ca.crt.' The timestamp should be updated to when the CA cert was uploaded. You can cat the file or run the below command to check the file:
   openssl x509 -in CA-chain.pem -noout -subject -issuer
4. View the issuers on the pkcs12 private/public bundle. Private Key Password needed.
   openssl pkcs12 -in hostname.site.ds.army.mil.pfx -nokeys | grep subject
5. Openssl command to check if certificate is in PEM format:
   openssl x509 -in cert.pem -text -noout
   a. If you get the following error it means that you are trying to view a non-PEM cert.
      unable to load certificate
      12626:error:0906D06C:PEM routines:PEM_read_bio:no start line:pem_lib.c:647:Expecting: TRUSTED CERTIFICATE

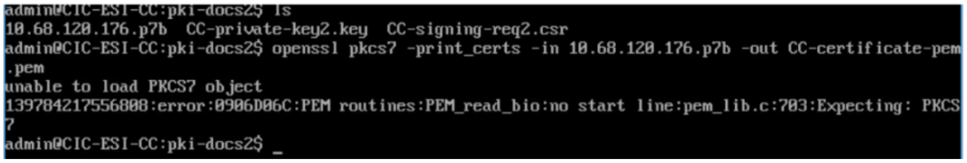
APPENDIX B: Common Errors

1. When Generating the pkcs 12 bundle for Server Authorization you may see this error:

   ```
   root@CIC-ESI-CC:admin# openssl pkcs12 -export -in 10.68.120.176.cer -inkey private.key -out newpriva
   tekey.pfx
   Enter pass phrase for private.key:
   unable to load certificates
   140648797476680:error:0D0680A8:asn1 encoding routines:ASN1_CHECK_TLEN:wrong tag:tasn_dec.c:1343:
   140648797476680:error:0D07803A:asn1 encoding routines:ASN1_ITEM_EX_D2I:nested asn1 error:tasn_dec.c:
   393:Type=X509_CINF
   140648797476680:error:0D08303A:asn1 encoding routines:ASN1_TEMPLATE_NOEXP_D2I:nested asn1 error:tasn
   _dec.c:777:Field=cert_info, Type=X509
   140648797476680:error:0907400D:PEM routines:PEM_X509_INFO_read_bio:ASN1 lib:pem_info.c:259:
   ```

   Please review the .key and .cer file for spaces or line returns. The CA .cer file is in the wrong format. Please confirm the .cer file is in PEM format
2. Passphrases with special characters:
   a. Special characters like exclamation points may cause problems since shell can misinterpret these characters. A workaround is to force input the passphrases into the openssl command. This will bypass the passphrase prompt.
   openssl pkcs12 -export -in CC-CA-public.cer -inkey*private.key -out cc-server.pfx *   **-name CNAME -passin 'pass:exp@ss!!!word' -passout 'pass:exp@ss!!!word'**
3. Pkcs7 to PEM conversion fails with below error:

   ```
   admin@CIC-ESI-CC:pki-docs2$ ls
   10.68.120.176.p7b  CC-private-key2.key  CC-signing-req2.csr
   admin@CIC-ESI-CC:pki-docs2$ openssl pkcs7 -print_certs -in 10.68.120.176.p7b -out CC-certificate-pem
   .pem
   unable to load PKCS7 object
   139784217556808:error:0906D06C:PEM routines:PEM_read_bio:no start line:pem_lib.c:703:Expecting: PKCS
   7
   admin@CIC-ESI-CC:pki-docs2$ _
   ```

   This can occur if the file uses DER cipher. Please use the openssl command to perform the file conversion:
   openssl pkcs7 -in cert.p7b -inform DER -print_certs -out cert.pem

APPENDIX C: Log Debugging

1. In the CLI type the below commands to turn on proper log debugging to view Certificate info:
   log level set DEBUG API com.lumeta.api.impl.SessionServiceImpl
   log level set DEBUG API com.lumeta.api.dao.UserDaoImpl
2. View the Lumeta-webapp.out for "Looking for DN" and match the DN column with the database command select * from sytem.user_certificate.