Generate and Add Certificates that are Required for Installation of Secure Endpoint Private Cloud 3.x Onwards

Contents

Introduction

This document describes the process to generate certificates that have to be uploaded with every fresh installation of Secure Console Private Cloud or to renew the installed Certificate Services.

Prerequisites

Requirements

The information in this document is based on these software and hardware versions:

- Windows Server 2008
- CentOS 7/8
- Secure Console Virtual Private Cloud 3.0.2 (Onwards)
- OpenSSL 1.1.1

Components Used

Cisco recommends that you have knowledge of these topics:

- Windows Server 2008 (Onwards)
- Secure Console Private Cloud installation
- Public Key Infrastructure
- OpenSSL
- Linux CLI

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

With the introduction of Secure Console Private Cloud 3.X, hostnames, and certificate/key pairs are required for all of the following services:

- Administration Portal
- Authentication (new in Private Cloud 3.X)
- Secure Console
- Disposition Server
- Disposition Server - Extended Protocol
- Disposition Update Service
- Firepower Management Center

This document is discussed a quick way to generate and upload the required certificates. You can tweak each of the parameters, including the hashing algorithm, key size, and others, as per your organization's policy, and your mechanism of generating these certificates might not match with what is detailed here.

> **Warning**: The procedure mentioned below can vary as per your CA server configuration. It is expected that the CA server of your choice is already provisioned and the configuration of the same has been completed. The following technote just describes an example of generating the certificates and Cisco TAC is not involved in troubleshooting issues related to certificate generation and/or CA server issues of any kind.

Certificate Creation

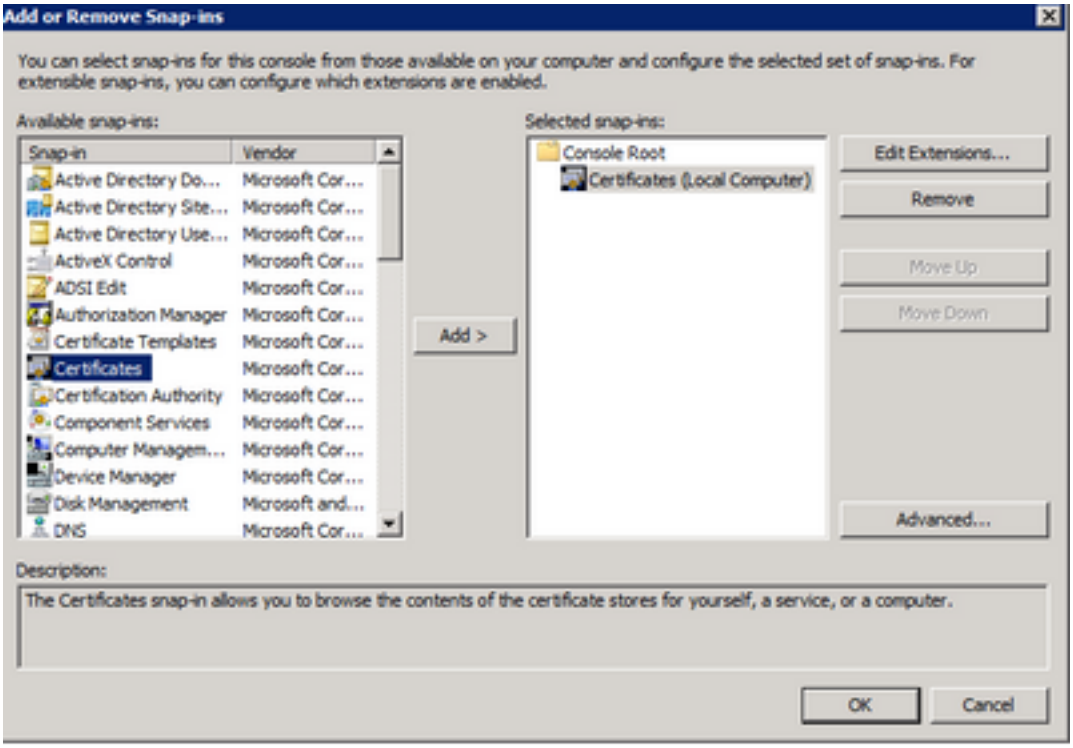
Generate Certificates on the Window Server

Ensure that the following roles are installed and configured on your Windows Server.

- Active Directory Certificate Services
- Certification Authority
- Certification Authority Web Enrollment
- Online Responder
- Certificate Enrollment Web Service
- Certificate Enrollment Policy Web Service
- Active Directory Domain Services
- DNS Servers
- Web Server (IIS)

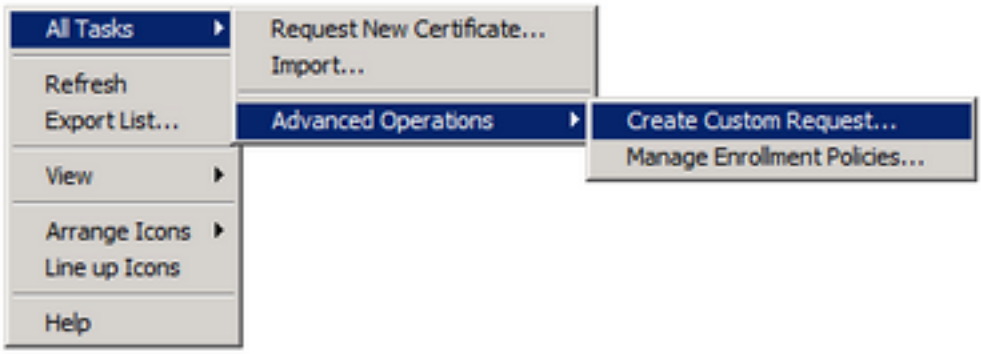
Generate a Certificate Signing Request (CSR)

Step 1. Navigate to MMC console, and add the Certificates snap-in for your computer account as shown in the image here.

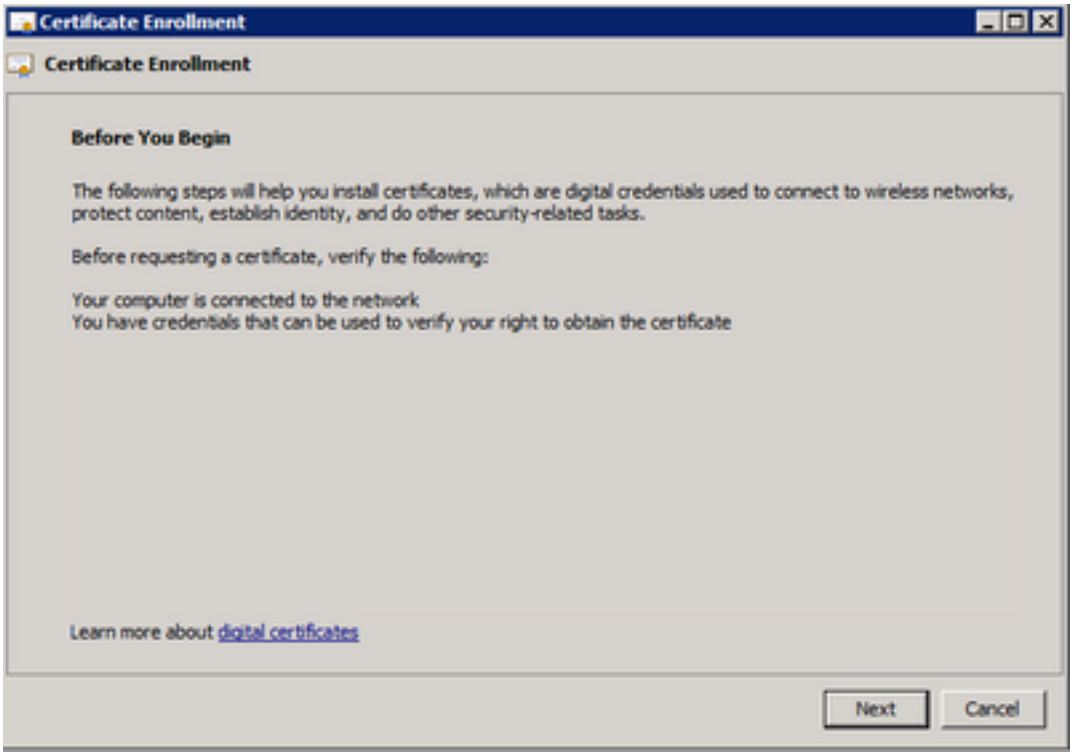


Step 2. Drill down **Certificates (Local Computer) > Personal > Certificates**.

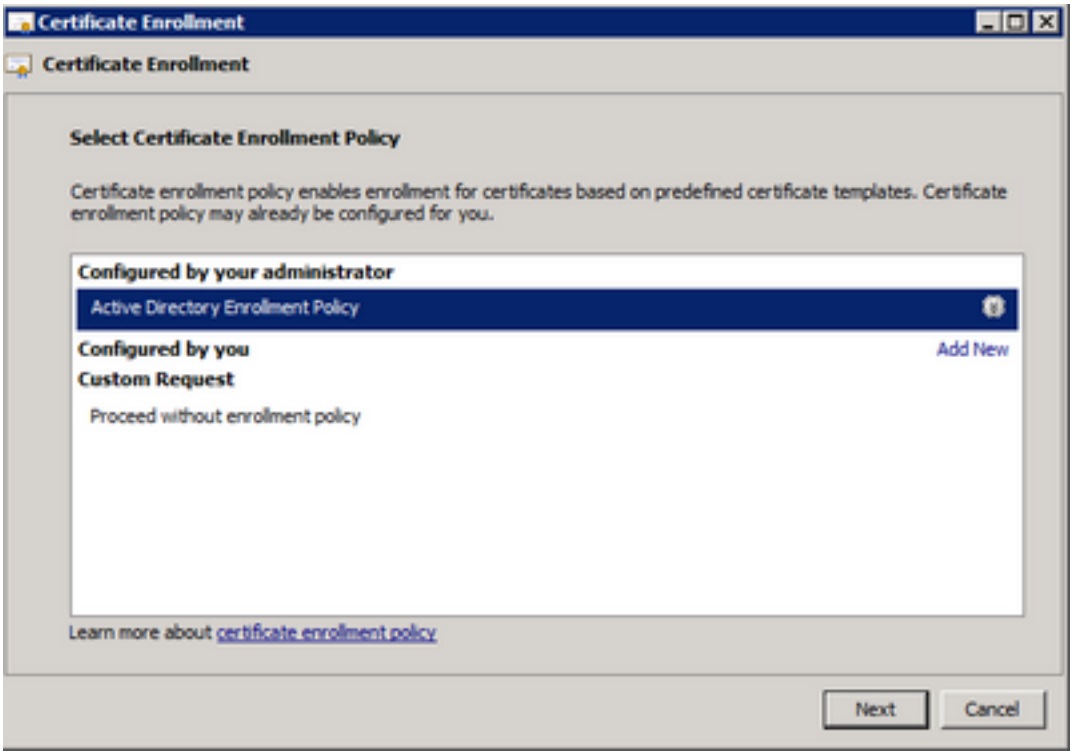
Step 3. Right-click on the empty space and select **All Tasks > Advanced Operations > Create Custom Request.**



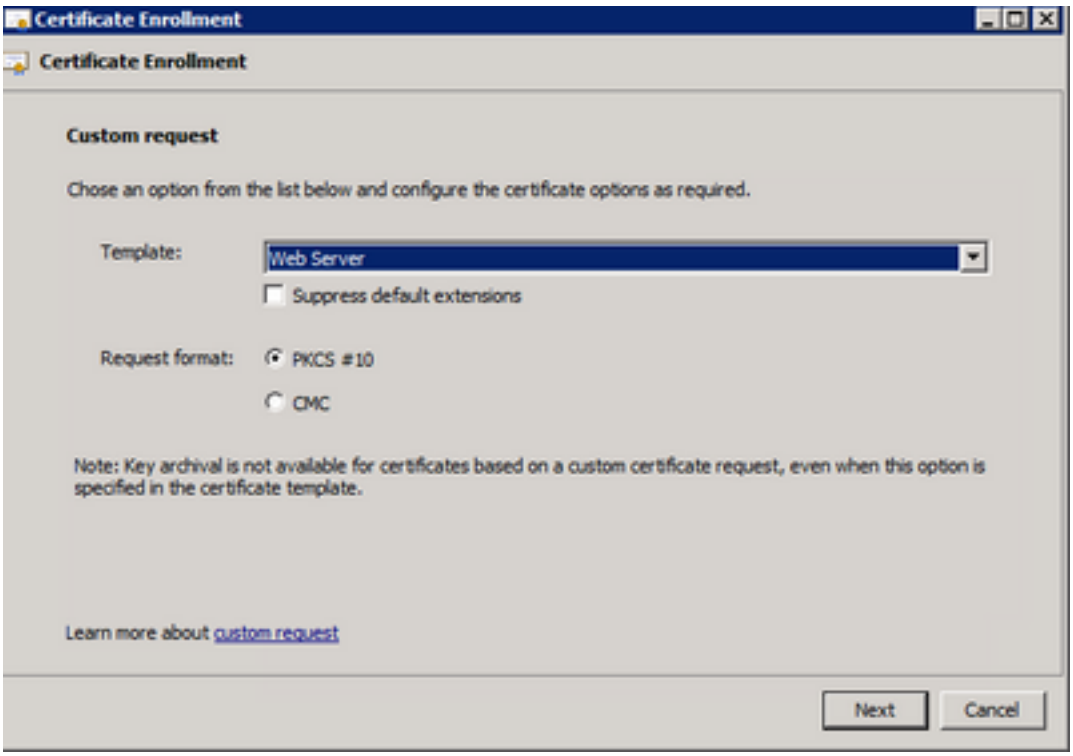
Step 4. Select **Next** at the Enrollment window.



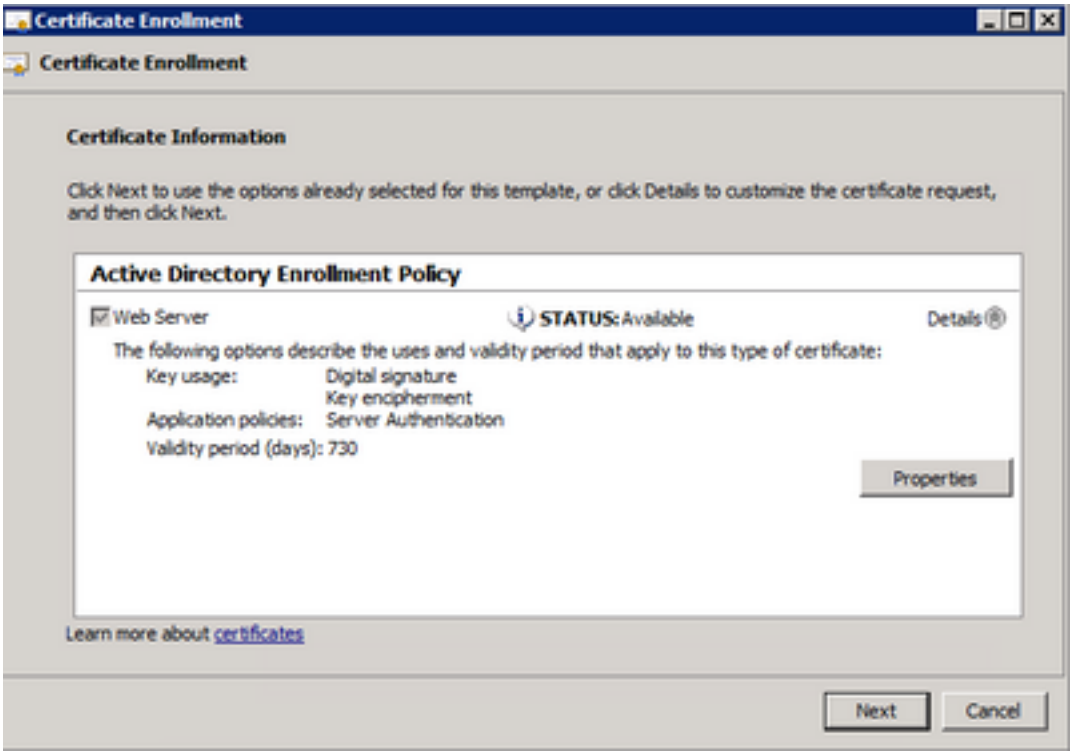
Step 5. Select your certificate enrollment policy and select **Next**.



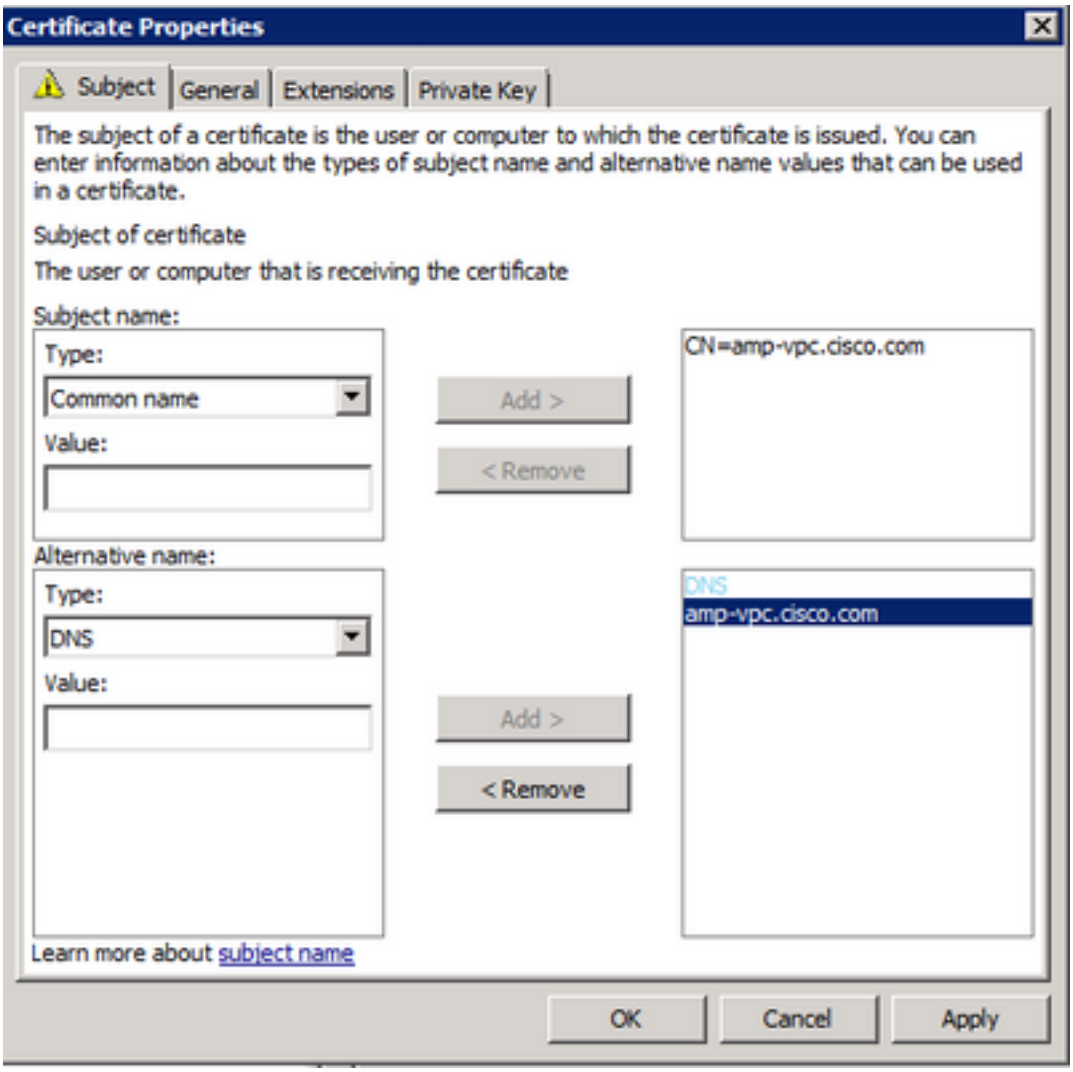
Step 6. Choose the template as **Web Server** and select **Next**.



Step 7. If your "Web Server" template has been configured correctly and is available for enrollment, the status Available is displayed. Select **Details** to expand Properties.

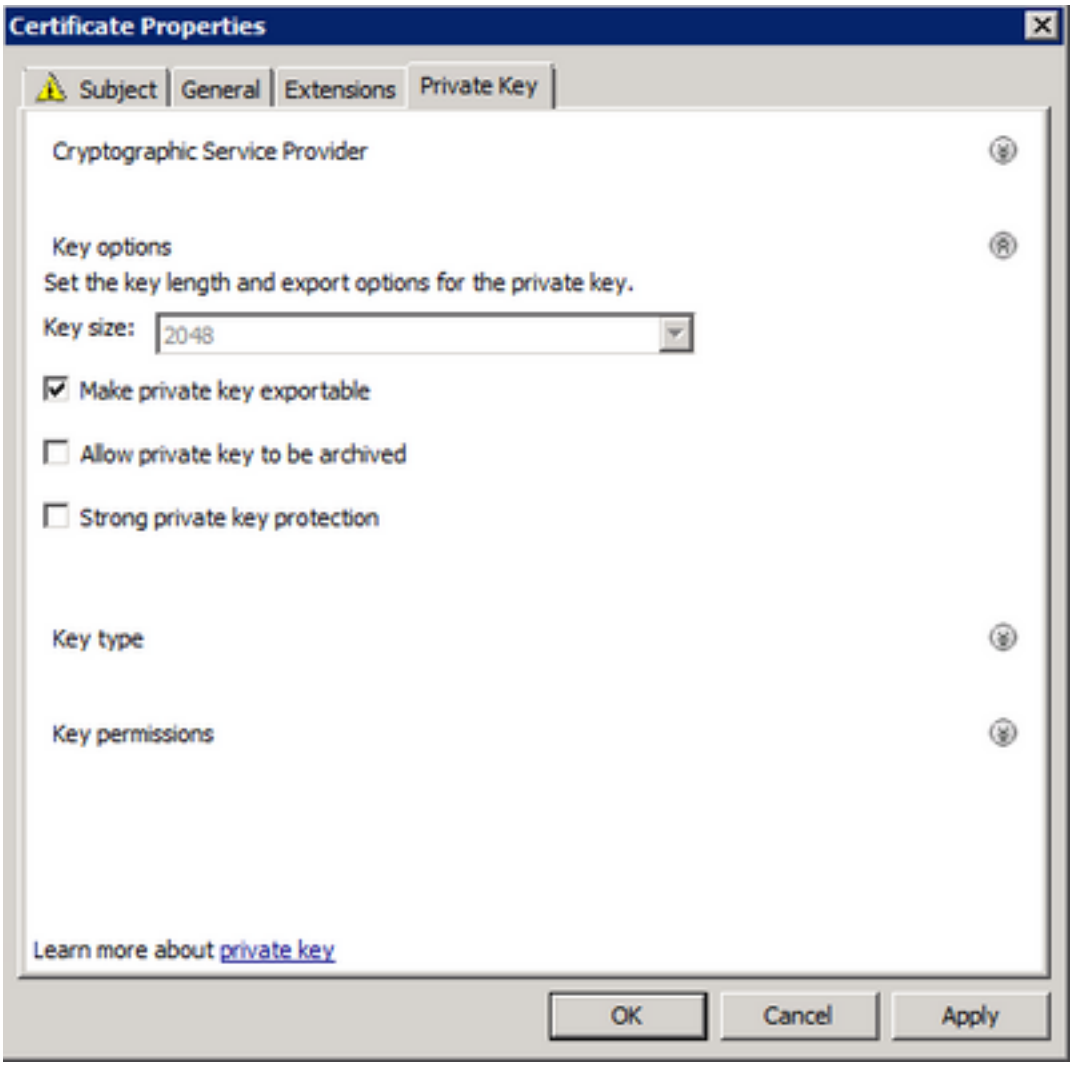


Step 8. At a minimum, add the CN and DNS attributes. The rest of the attributes can be added as per your security requirements.



Step 9. Optionally, give a Friendly Name under the **General** tab.

Step 10. Select on the **Private Key** tab and ensure that you're enabling **Make private key exportable** under the **Key Options** section.

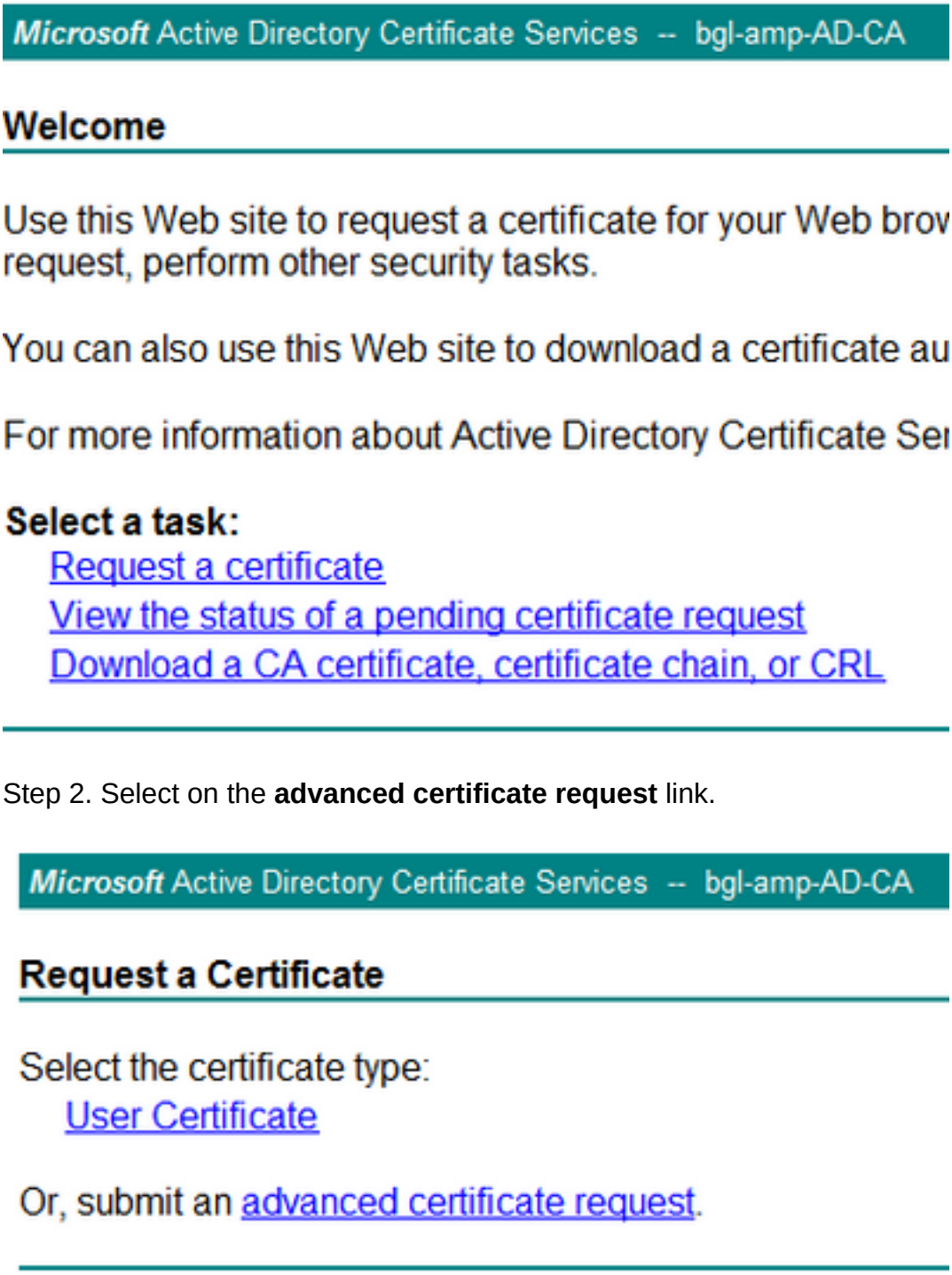


Step 11. Finally, select on **OK**. This must lead you to the Certificate Enrollment dialog from where you can select **Next**.

Step 12. Browse to a location to save the .req file which is submitted to the CA server for signing.

Submitting the CSR to the CA and generating the certificate

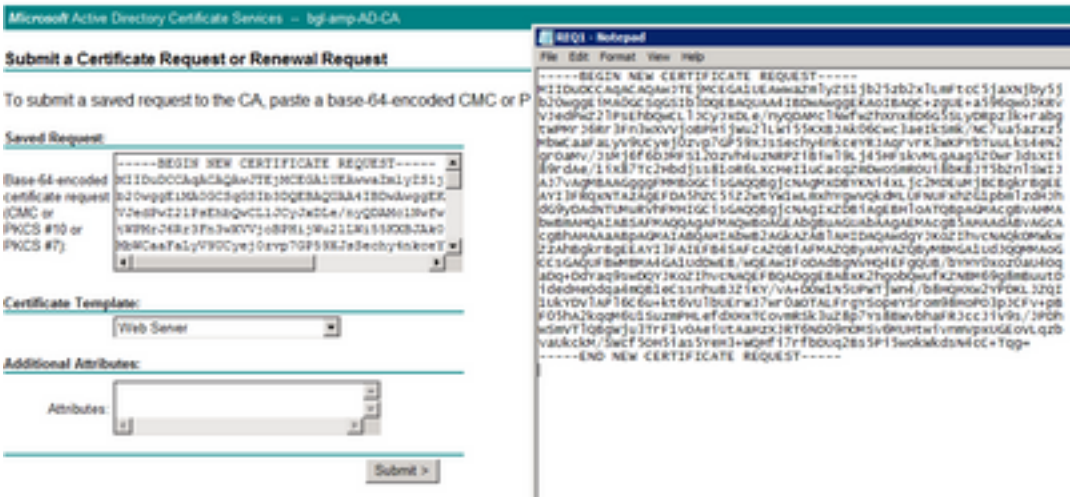
Step 1. Navigate to your MS AD Certificate Services Web Page as below and select **Request a Certificate**.



Step 2. Select on the **advanced certificate request** link.

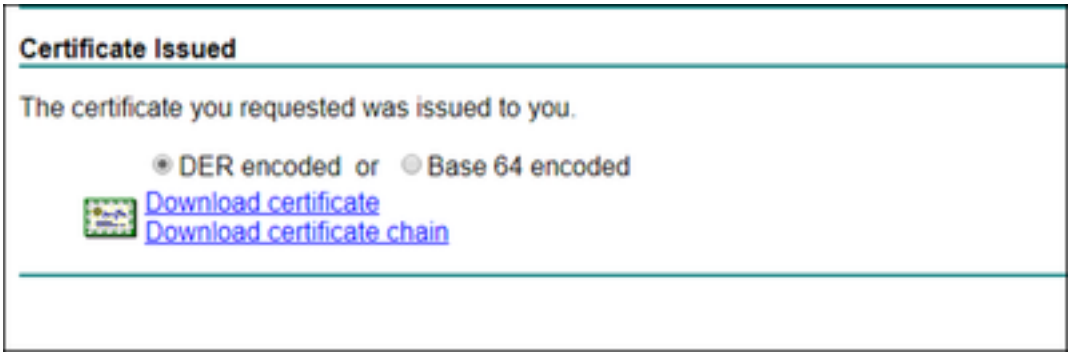
Step 3. Select on **Submit a certificate request by using a base-64-encoded CMC or PKCS #10 file, or submit a renewal request by using a base-64-encoded PKCS #7 file**.

Step 4. Open the contents of the previously saved .req file (CSR) via Notepad. Copy the contents and paste it here. Ensure that the Certificate Template is selected as **Web Server**



Step 5. Finally, select **Submit**.

Step 6. At this point, you must be able to **Download** the certificate, as shown in the image.



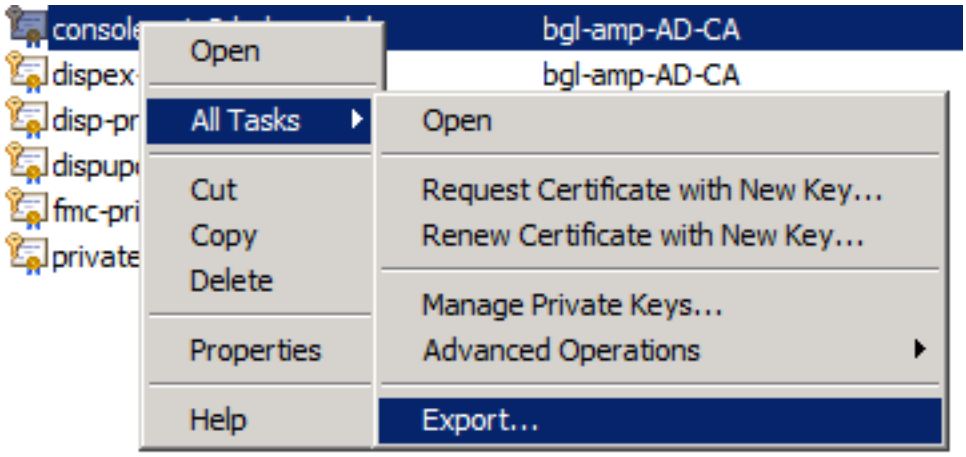
Exporting the Private Key and converting to PEM format

Step 1. Install the certificate into your Certificate Store by opening the .cer file and select **Install Certificate**.

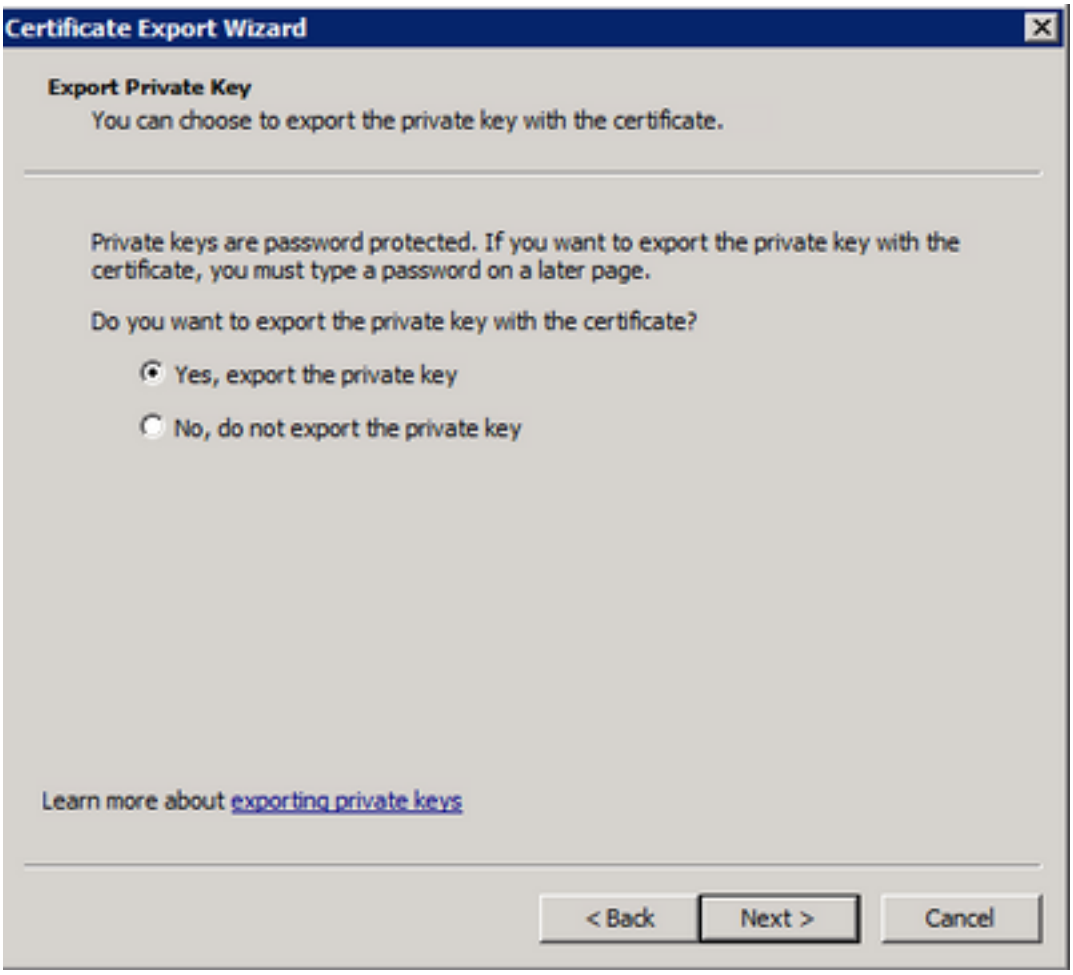
Step 2. Navigate to the MMC snap-in that was selected earlier.

Step 3. Navigate to the store where the certificate was installed.

Step 4. Right-click the correct certificate, select **All Tasks > Export**.



Step 5. At the Certificate Export Wizard, confirm to export the private key, as shown in the image.



Step 6. Enter a password and select **Next** to save the private key on your disk.

Step 7. This saves the private key in .PFX format, however, this needs to be converted to .PEM format to use this with Secure Endpoint Private Cloud.

Step 8. Install OpenSSL libraries.

Step 9. Open a command prompt window and change to the directory where you installed OpenSSL.

Step 10. Run the following command to extract the private key and save it to a new file: (If your PFX file is not in the same path as where the OpenSSL library is stored, you have to specify the exact path along with the filename)

```
openssl pkcs12 -in yourpfxfile.pfx -nocerts -out privatekey.pem -nodes
```

Step 11. Now run the following command to also extract the public cert and save it to a new file:

```
openssl pkcs12 -in yourpfxfile.pfx -nokeys -out publiccert.pem -nodes
```

Generate Certificate on Linux Server (Strict SSL check DISABLED)

Note: Strict TLS Check verifies that the certificate meets Apple's TLS requirements. Please refer to the Admin Guide for more information.

Ensure that the Linux Server that you're trying to generate the required certificates has the OpenSSL 1.1.1 libraries installed. Verifying if this and the procedure listed below can vary from the Linux distribution that you're running. This portion has been documented, as done on a CentOS 8.4 Server.

Generate Self Signed RootCA

Step 1. Generate the Private Key for Root CA certificate.

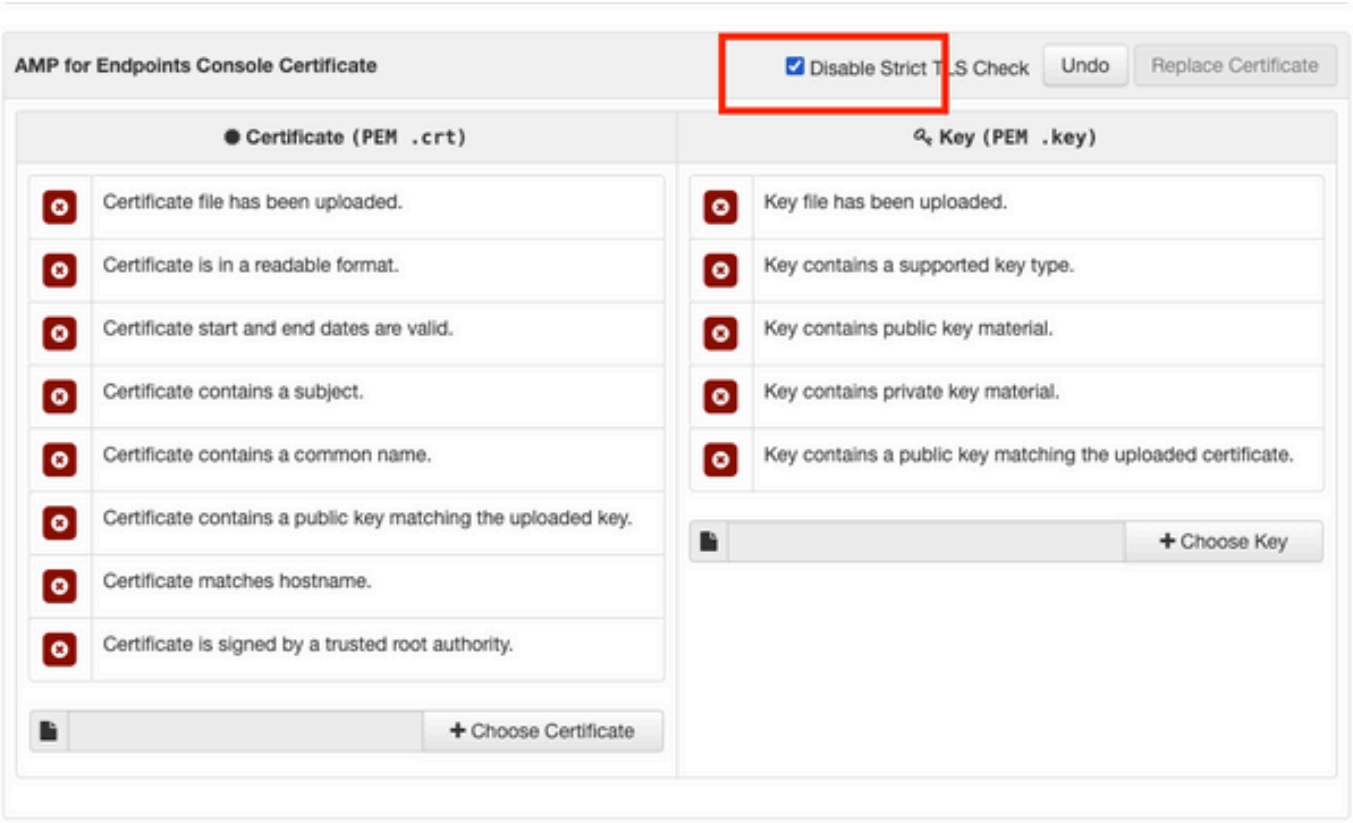
```
openssl genrsa -out <YourRootCAName.key> 4096
```

Step 2. Generate the CA certificate.

```
openssl req \
-subj '/CN=<YourRootCAName>/C=US/OU=<YourDepartmentName>/O=<YourCompanyName>' \
-addext "extendedKeyUsage = serverAuth, clientAuth" \
-outform pem -out <YourRootCAName.pem> \
-key <YourRootCAName.key> -new -x509 \
-days "1000"
```

Generate a certificate for each service

Create the certificate for Authentication, Console, Disposition, Disposition-Extended, Update server, Firepower Management Center(FMC) service as per the DNS name entry. You need to repeat the below certificate generate process for each service (Authentication, Console etc.).



Generate Private key

```
openssl genrsa -out <YourServiceName.key> 4096
```
Replace the <YourServiceName.key> with the new KEY filename to be created as Auth-Cert.key

Generate CSR

```
openssl req -new \
-subj '/CN=<YourServiceName>/C=US/OU=<YourDeptName>/O=<YourCompanyName>' \
-key <YourServiceName.key> -out <YourServiceName.csr>
```
Replace the <YourServiceName.key> with the current (or new) certificate KEY file such as Auth-Cert.key

Replace the <YourServiceName.csr> with CSR filename to be created such as Auth-Cert.crt

Generate Certificate

```
openssl x509 -req \
-in <YourServiceName.csr> -CA <YourRootCAName.pem> \
-CAkey <YourRootCAName.key> -CAcreateserial -out <YourServiceName.crt> \
-days 397 -sha256
```
Replace the <YourServiceName.csr> with actual (or new) certificate CSR such as Auth-Cert.csr

Replace the <YourRootCAName.pem> with actual (or new) PEM filename as RootCAName.pem

Replace the <YourServiceName.key> with the current (or new) certificate KEY file such as Auth-Cert.key

Replace the <YourServiceName.crt> with filename to be created such as Auth-Cert.crt

Generate Certificate on Linux Server (Strict SSL check ENABLED)

> **Note**: Strict TLS Check verifies that the certificate meets Apple's TLS requirements. Please refer to the Admin Guide for more information.

Generate Self Signed RootCA

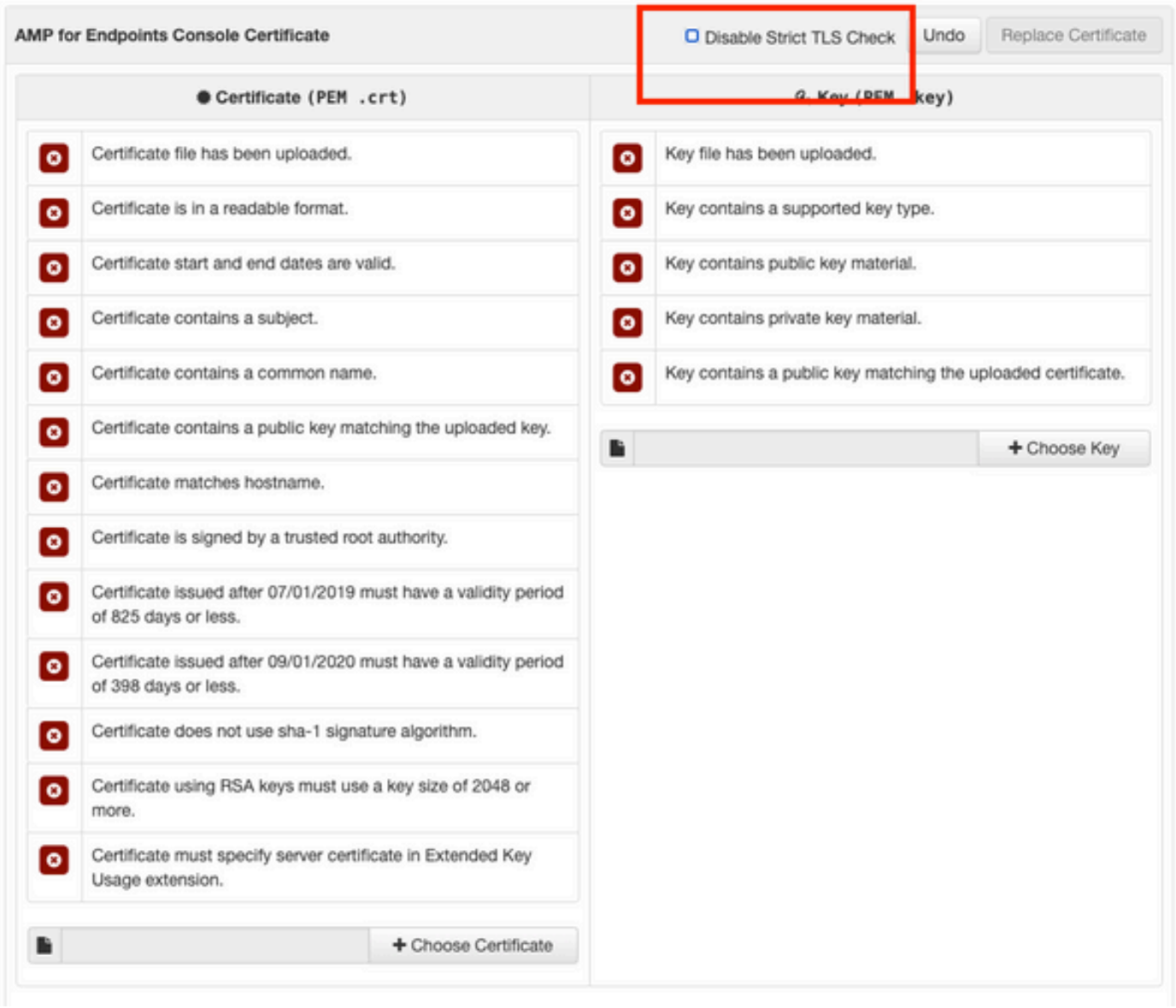
Step 1. Generate the Private Key for Root CA certificate.

```
openssl genrsa -out <YourRootCAName.key> 4096
```
Step 2. Generate the CA certificate.

```
openssl req \
-subj '/CN=<YourRootCAName>/C=US/OU=<YourDepartmentName>/O=<YourCompanyName>' \
-outform pem -out <YourRootCAName.pem> \
-key <YourRootCAName.key> -new -x509 \
-days "1000"
```

Generate a certificate for each service

Create the certificate for Authentication, Console, Disposition, Disposition-Extended, Update server, Firepower Management Center(FMC) service as per the DNS name entry. You need to repeat the below certificate generate process for each service (Authentication, Console etc.).



Create an Extensions Configuration file and save it (extensions.cnf)

```
[v3_ca]
basicConstraints = CA:FALSE
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = critical, serverAuth, clientAuth
```

Generate Private key

```
openssl genrsa -out <YourServiceName.key> 4096
```

Replace the <YourServiceName.key> with a new KEY filename to be created as Auth-Cert.key

Generate CSR

```
openssl req -new \
-key <YourServiceName.key> \
-subj '/CN=<YourServiceName>/C=US/OU=<YourDeptName>/O=<YourCompanyName>' \
-out <YourServiceName.csr>
```

Replace the <YourServiceName.key> with the current (or new) certificate KEY such as Auth-Cert.key

Replace the <YourServiceName.csr> with the current (or new) certificate CSR such as Auth-Cert.csr

Generate Certificate

```
openssl x509 -req -in <YourServiceName.csr> \
-CA <YourRootCAName.pem> -CAkey <YourRootCAName.key> \
-CAcreateserial -out <YourServiceName.crt> \
-extensions v3_ca -extfile extensions.cnf \
-days 397 -sha256
```

Replace the <YourServiceName.csr> with current (or new) certificate CSR such as Auth-Cert.csr

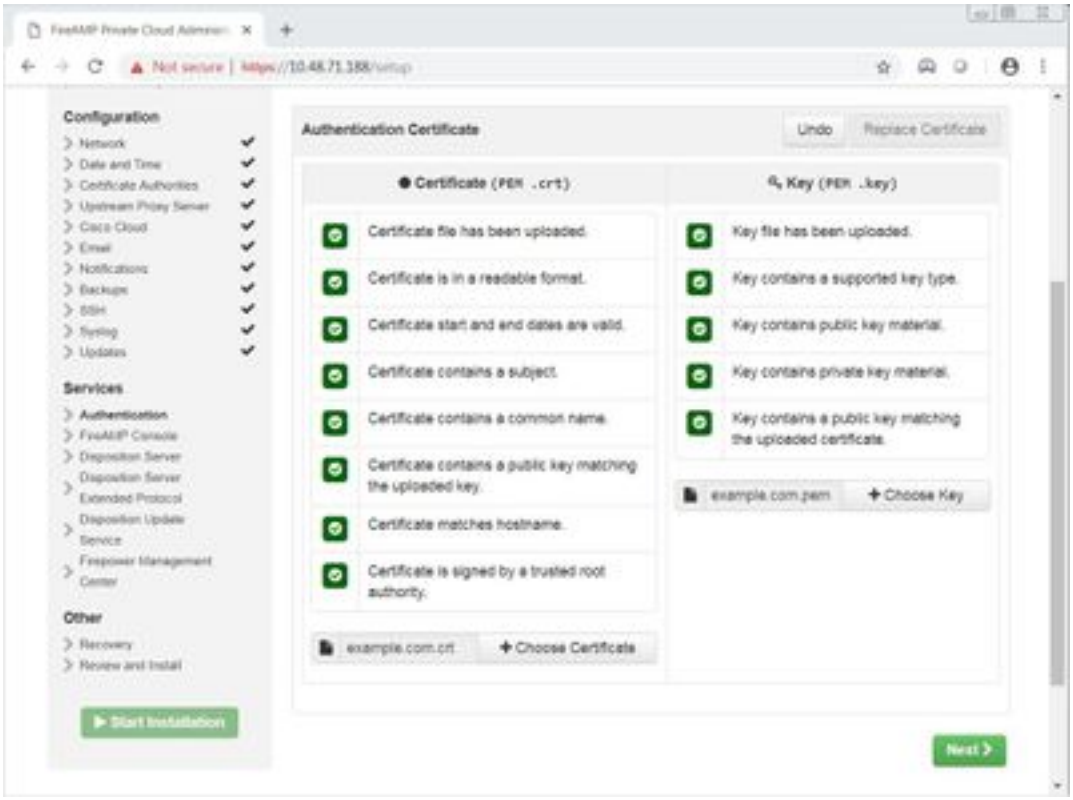
Replace the <YourRootCAName.pem> with current (or new) PEM filename as RootCAName.pem

Replace the <YourServiceName.key> with current (or new) certificate KEY file such as Auth-Cert.key

Replace the <YourServiceName.crt> with filename to be created such as Auth-Cert.crt

Adding The Certificates to Secure Console Private Cloud

Step 1. Once the certificates are generated from any of the above methods, upload the corresponding certificate for each of the services. If they have been generated correctly, all the checkmarks are enabled as seen in the image here.



Verify

There is currently no verification procedure available for this configuration.

Troubleshoot

There is currently no specific troubleshooting information available for this configuration.