Avaya Solution & Interoperability Test Lab

Configuring Avaya Aura® System Manager 6.2 FP2 and Avaya Aura® Session Manager 6.2 FP2 to use Third-Party Security Certificates for Transport Layer Security - Issue 1.1

Abstract

These Application Notes describe the steps to configure Avaya Aura® System Manager 6.2 Feature Pack 2 and Avaya Aura® Session Manager 6.2 Feature Pack 2 to use Transport Layer Security and certificates signed by a customer or third-party Certification Authority. The default Avaya product identification certificates and Avaya trusted root certificates are replaced with certificates signed by customers own Certification Authority servers or by a third-party Certificate Authority. These Application Notes are intended for customers who intend to replace default Avaya supplied certificates in a high security networked environment, and who wish to secure signaling.

Information in these Application Notes has been obtained through Solution Integration compliance testing and additional technical discussions. Testing was conducted at the Avaya Solution and Interoperability Test Lab.

Table of Contents

1. Introduction 3
2. Interoperability Testing 3
   2.1. Test Results and Observations 4
3. Reference Configuration 5
4. Equipment and Software Validated 6
5. Configure Certificate Authority on Microsoft Windows Server 2008 R2 Enterprise 7
   5.1. Add Active Directory Certificate Services Role 7
   5.2. Create a Certificate Template 10
6. Configure Certificates for Avaya Aura® System Manager 14
   6.1. Generate a Certificate Signing Request and Private Key for Avaya Aura® System Manager 14
      6.1.1. Edit the OpenSSL Default Configuration File 14
      6.1.2. Generate the CSR and Private Key for System Manager 16
   6.2. Process Certificate Signing Request on Certificate Authority 17
   6.3. Package the Private Key and Signed Certificate in a PKCS#12 format 19
   6.4. Install the trusted Root CA certificate in Avaya Aura® System Manager 20
   6.5. Replace Avaya Aura® System Manager Identity Certificate 22
7. Configure Certificates for Avaya Aura® Session Manager 24
   7.1. Install third party Trusted Root CA Certificate on Avaya Aura® Session Manager 24
   7.2. Restart Avaya Aura® System Manager JBoss Service 25
   7.3. Generate a Certificate Signing Request and Private Key for Avaya Aura® Session Manager 25
      7.3.1. Edit the default OpenSSL Configuration File 26
      7.3.2. Generate a CSR and Private Key 27
   7.4. Process the Certificate Signing Request on Certificate Authority 29
   7.5. Package the Session Manager Private key and Signed Certificate in PKCS#12 format 30
   7.6. Replace the Default Avaya Aura® Session Manager Identity Certificate 30
8. Verification Steps 34
   8.1. Avaya Aura® System Manager Verification 34
   8.2. Avaya Aura® Session Manager Verification 37
9. Conclusion 43
10. Additional References 43

1. Introduction

These Application Notes describe the configuration of Avaya Aura® System Manager 6.2 Feature Pack (FP) 2 and Avaya Aura® Session Manager 6.2 FP2 with Transport Layer Security (TLS) using third-party Certificate Authority (CA) certificates. Digital Certificates certify that a public key belongs to its reputed owner. To ensure greater trust, a trusted party can sign the public key and the information about its owner, creating a public-key certificate. The certificate guarantees the identity of its bearer. A trusted party that issues digital certificates is called a Certification Authority (CA). A CA can be a third-party external service provider, e.g., VeriSign or Entrust, or belong to the same organization as the entity it serves.

In the context of Avaya Aura® System Manager and Avaya Aura® Session Manager, the certificate that is used to assert its identity is called a product certificate or an identity certificate. The issuer or CA certificate used by Avaya Aura® System Manager and Avaya Aura® Session Manager to verify and validate the identity of the far end is referred to as the trusted certificate or root CA certificate.

TLS sessions use a client-server model. Clients (i.e., devices requiring a service) contact a server and are offered an identity certificate as proof of the server's integrity. Clients verify the offered certificate by testing authenticity with a common trusted root CA certificate. If successfully authenticated; the client and server commence negotiations on an encryption scheme, and if successful, transmission is secured from that point. TLS protocol allows for servers to request a certificate from a client and will authenticate it using a trusted root CA certificate. This is known as mutual authentication and is preferable to one-way authentication as it prevents unauthorized hosts from obtaining services. Avaya Aura® Session Manager uses mutual authentication. Servers can only offer one identity certificate, but may have several trusted root CA certificates.

Non-unique, default TLS certificates, certified by Avaya, are shipped with Avaya Aura® System Manager and Avaya Aura® Session Manager to provide out-of-box support for TLS sessions. For production environments, Avaya recommends replacing these default certificates with customer CA or third-party CA signed unique identity certificates. These Application Notes describe how to replace default certificates with certificates signed by a third-party certification service. It is assumed that both Avaya Aura® System Manager and Avaya Aura® Session Manager are already installed, configured, and operational.

2. Interoperability Testing

These Application Notes focus on replacing System Manager default identity certificate for container TLS service, Apache HTTP Service and Management Service. For Session Manager, the default identity certificate is replaced for SIP Security Module, used for securing SIP telephony communications, Security Module HTTPS, and the Management Service.

Testing was completed to ensure the new third-party certificates were being used to secure SIP traffic to Session Manager and management interactions with System Manager.

2.1. Test Results and Observations

All test cases were successful.

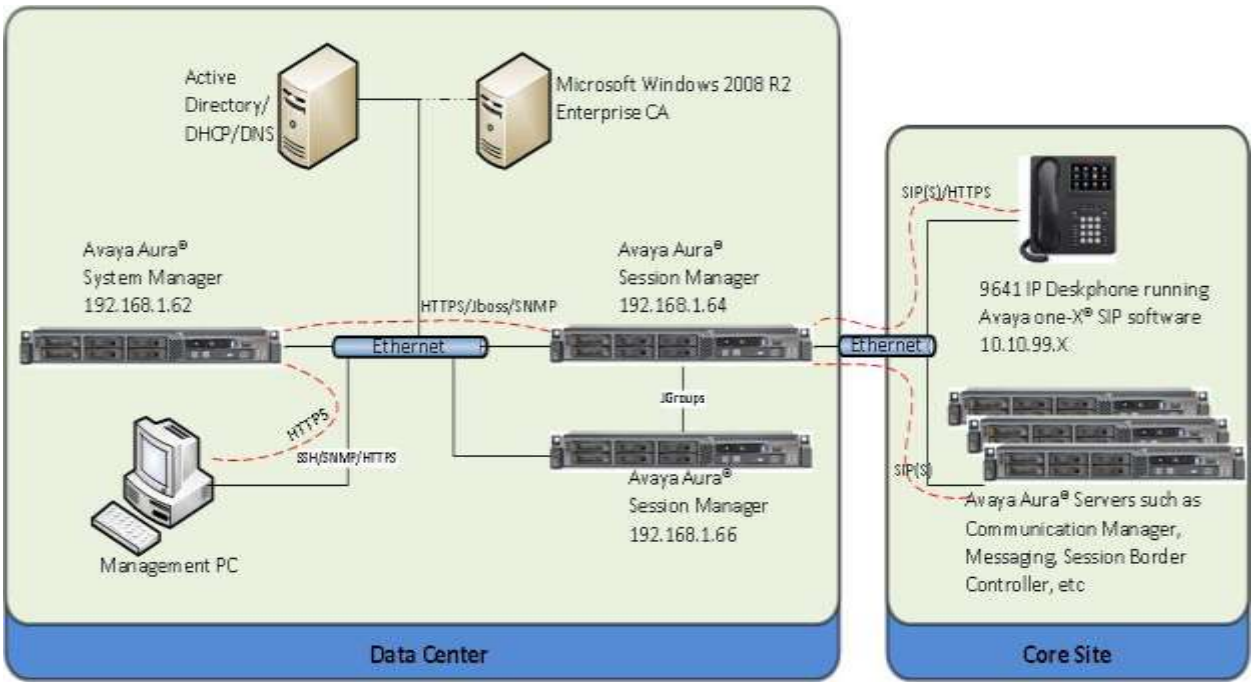
When adding a custom certificate template on Microsoft Windows 2008 Server, it was necessary to configure the template to support a minimum of Windows 2003 Server; otherwise the template would not show up on the Web Certificate Services menu.

Domain Name Server (DNS) verification is not performed on the certificate common name (CN) or Subject Alternative Name (SAN) using the configuration described in these Application Notes.

3. Reference Configuration

In the sample configuration shown in **Figure 1**, a standalone Avaya Aura® System Manager is used to manage two Avaya Aura® Session Manager elements. The second Session Manager can serve as a backup for the first Session Manager, in the case of a network or Session Manager failure. Microsoft Windows Server 2008 R2 Enterprise, deployed with Active Directory Certificate Services is used as a Certificate Authority. The CA is configured to generate certificates to use RSA public-key cryptography algorithm, 2048 bit key length and SHA1 hash algorithm. This CA can reside in the customer network or may reside at the third-party service provider data center.

System Manager common console contains a JBoss Application Server (AS) to manage various Avaya products including Session Manager. System Manager JBoss AS contains a JBoss Web Server where data is exchanged with Session Manager Management Interface over HTTPS. A JGroups Channel is established over a TCP connection between multiple Session Managers in the JBoss cluster and this is used for database replication and synchronization. Any Personal Computer (PC) on the network can access System Manager web console using HTTPS for administration purposes. The PC can also access System Manager or Session Manager servers via Secure Shell (SSH). 9641 IP Deskphones and Avaya Aura® Servers in the core network communicate with Session Manager using SIP. 9641 IP Deskphones running Avaya one-X® Deskphone SIP software, also use HTTPS to Session Manager for Personal Profile Manager (PPM) data. Each of the SIP or HTTP connections to System Manager and Session Manager are secured using TLS. TLS uses client and server authentication with X.509 public-key certificates obtained from the CA.



4. Equipment and Software Validated

The following equipment and software were used for the reference configuration

Equipment/Software	Release/Version
Avaya Aura® System Manager on Avaya S8800 Server	Release 6.2 FP2 Version: 6.3.2.4.1399
Avaya Aura® Session Manager on Avaya S8800 Server	Release 6.2 FP2 (6.3.2) Build 6.3.2.0.632023
Avaya 96x1 Series IP Deskphone (with Avaya one-X® SIP firmware)	Release 6.2.2.25 Build: 96x1-IPT-SIP-R6_2_2-060613
Hewlett Packard Compaq 6000 Pro Microtower PC	Microsoft Windows Server 2008 R2 Enterprise SP1 x64 • Active Directory Certificate Services Role

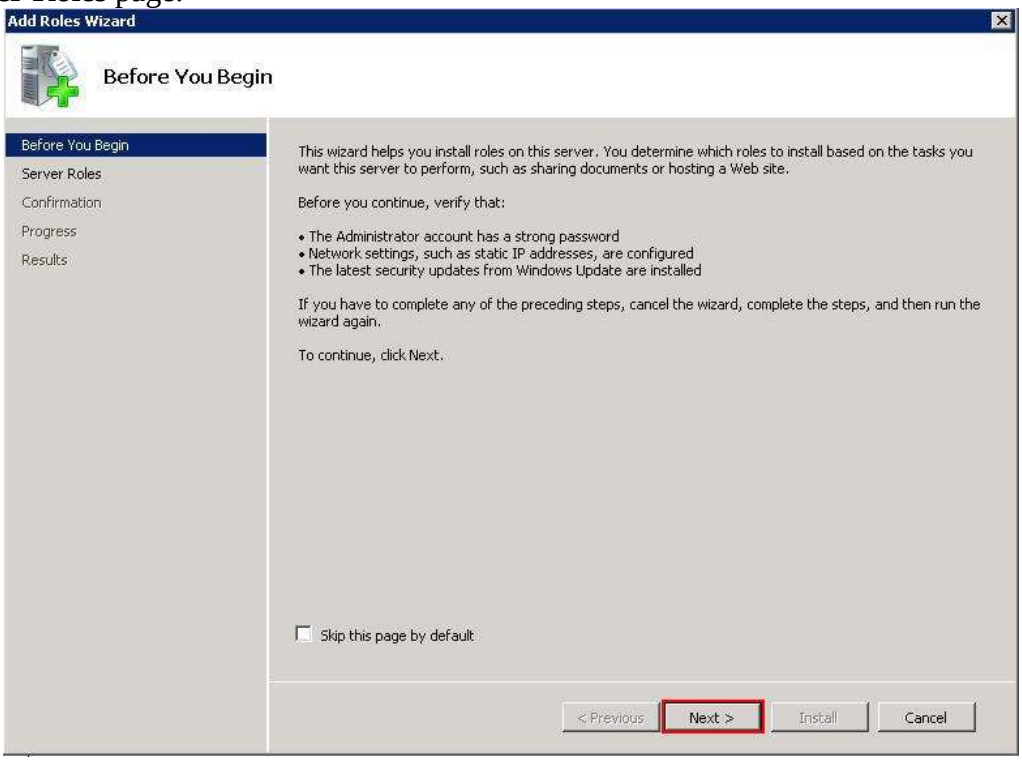
5. Configure Certificate Authority on Microsoft Windows Server 2008 R2 Enterprise

The digital certificates generated by a Certificate Authority (CA), certifies ownership of a public key by the named subject of the certificate. This allows other servers to rely upon signatures or assertions made by the private key that corresponds to the certified public key. The CA is trusted by both the subject (owner) of the certificate and the party relying on the certificate. A company may use a commercial CA which charge a fee to issue certificates. However many internet browsers and email clients may include a trusted certificate for this commercial CA, e.g., Verisign or Geotrust. A company may decide to use their own internal CA server and in this example, a Microsoft Windows Server 2008 R2 Enterprise is configured for Certificate Services and is used as the CA for the sample network.

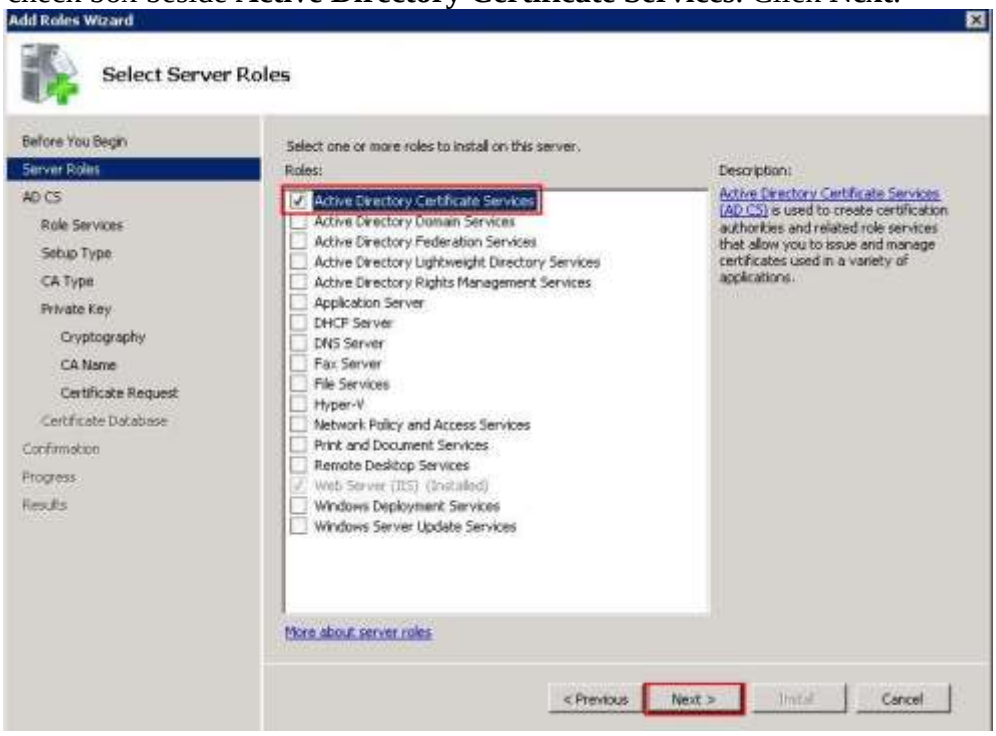
5.1. Add Active Directory Certificate Services Role

It is assumed that Microsoft Windows Server 2008 R2 Enterprise Edition is installed and working on a server within the network domain. Information is available from http://technet.microsoft.com for details on how to create or extend a Public-Key infrastructure (PKI). A PKI that meets the requirements of most organizations is a multi-tier CA with an off-line Root CA. In this example network, a single-tier standalone CA is used. This section details how to add and configure Microsoft Active Directory Certificate Services as a role on the Windows Server.

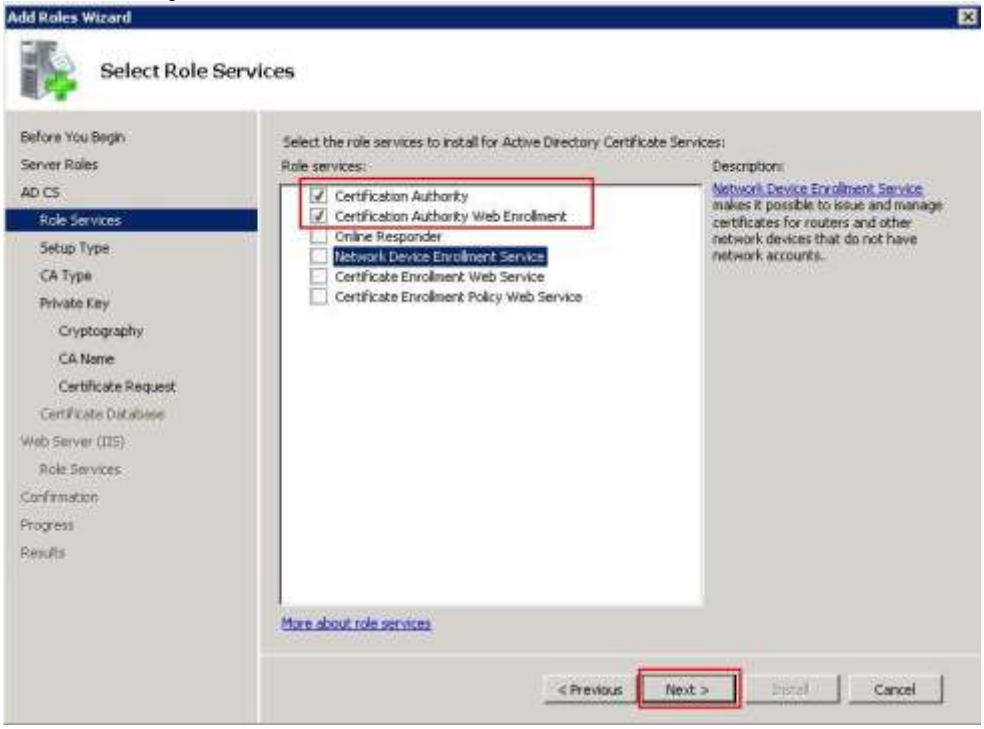
Log on to the CA server as administrator. Start the **Add Roles Wizard**. Select **Next** to move to the **Server Roles** page.



Select the check box beside **Active Directory Certificate Services**. Click **Next**.

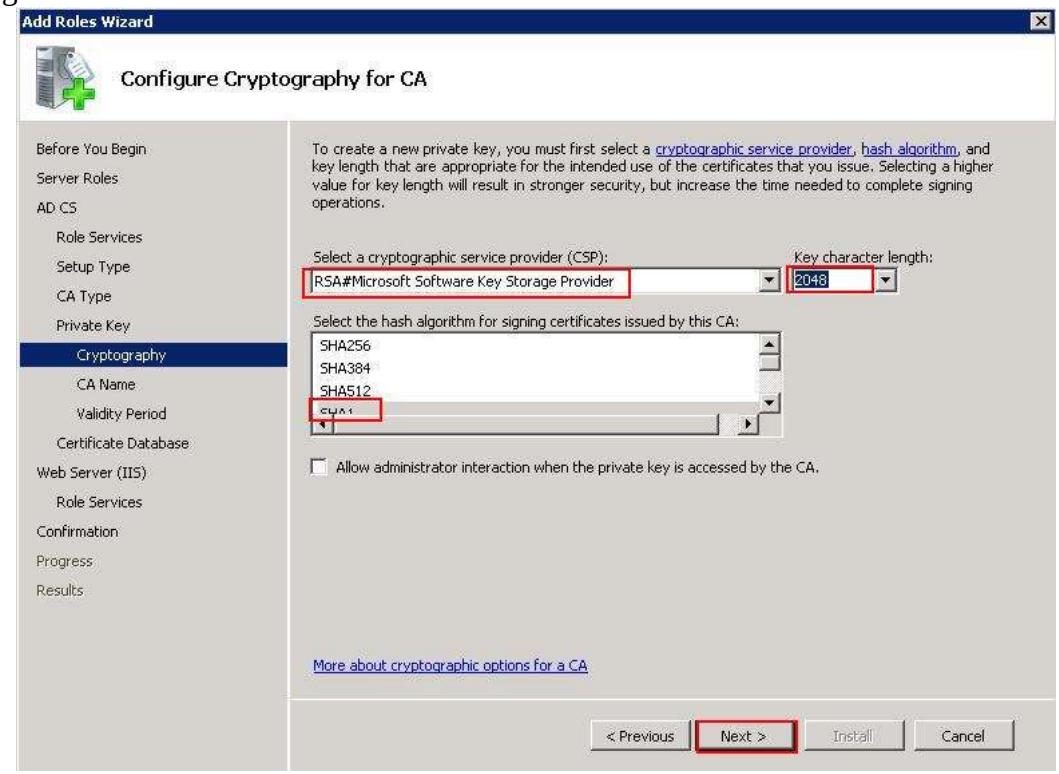


Click **Next**. (Not Shown) Select the check-box beside **Certification Authority** and **Certification Authority Web Enrollment**. Click **Next.**

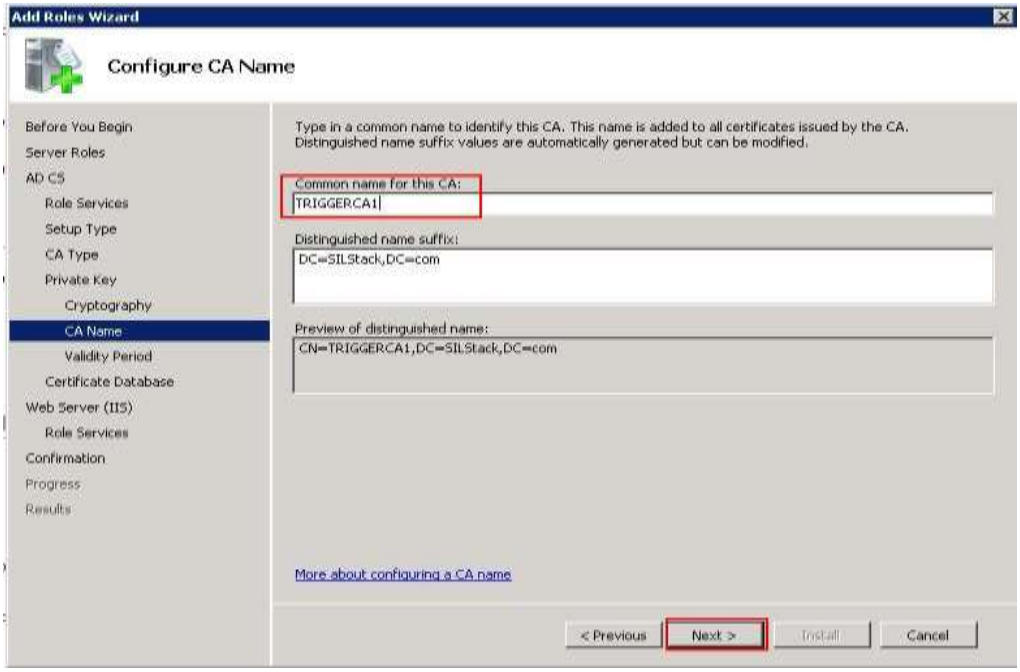


For the **Setup Type**, select **Enterprise** (Not Shown). Click **Next.** Select **Root CA** (Not shown) for the **CA Type.** Select **Create Private Key** (Not Shown). Click **Next.**

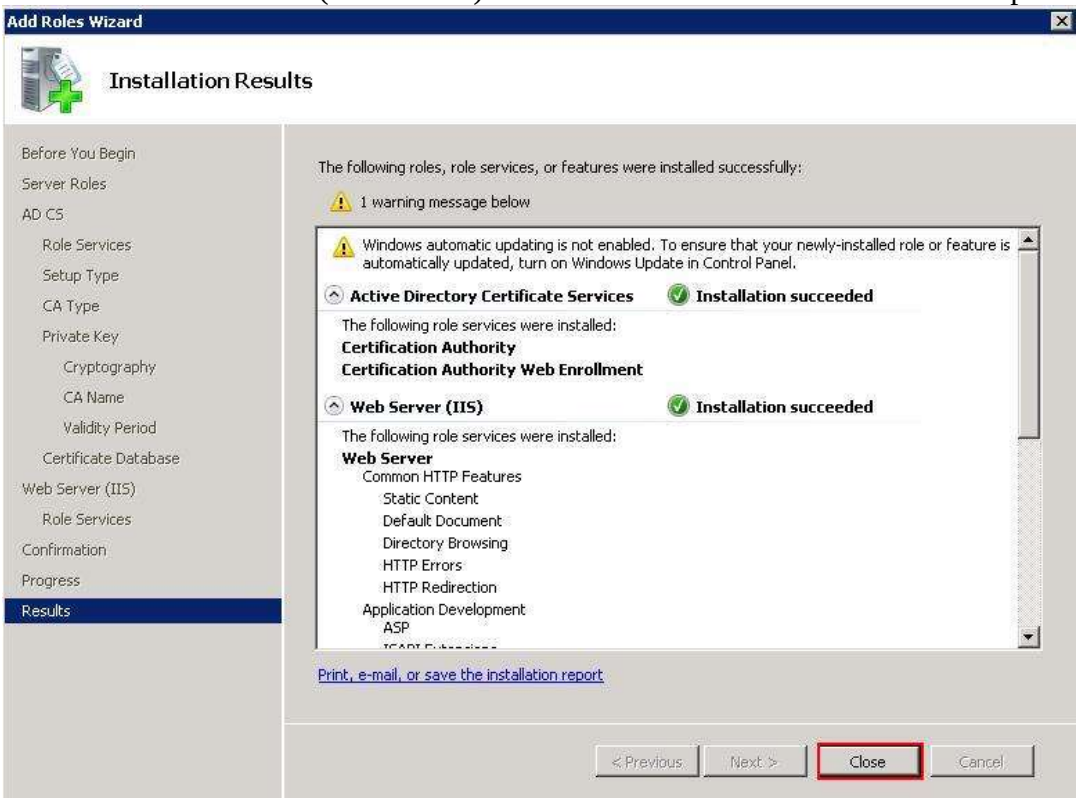
From the drop-down menu, select **RSA#Microsoft Software Key Storage Provider** as the cryptographic service provider. Select **2048** as the **Key character length**. Select **SHA1** as the hash algorithm. Click **Next**.



Enter a **Common name** for the CA. The **Distinguished Name Suffix** is optional. In this sample, components from the Active Directory domain name were used, e.g., DC=SILStack, DC=com. Click **Next**.



Use the default period of **5 years**. Click **Next.** Accept the default for **Certificate**, **IIS** and **Role Services** screens. Click **Install** (Not Shown). Click **Close** once the installation is complete.



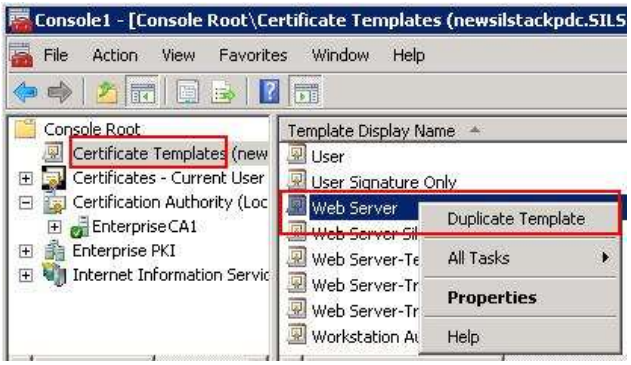
5.2. Create a Certificate Template

A new certificate template will be created to enforce a minimum key size of 2048 bits.
Once created, the certificate templates are automatically stored on the Active Directory Domain Controller.

1. On the Microsoft Windows Server, click **Start** and **Run**. Enter **MMC** and click **OK**.

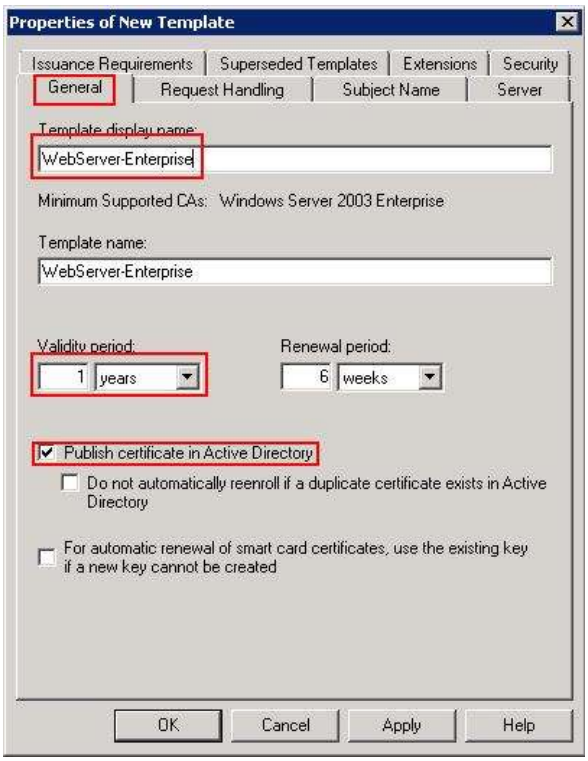
2. Click **File** and **Add/Remove Snap-In**. Select **Certificate Templates** and click **Add** and **OK**.

3. Right-click on the default **Web Server** template and select **Duplicate Template**.

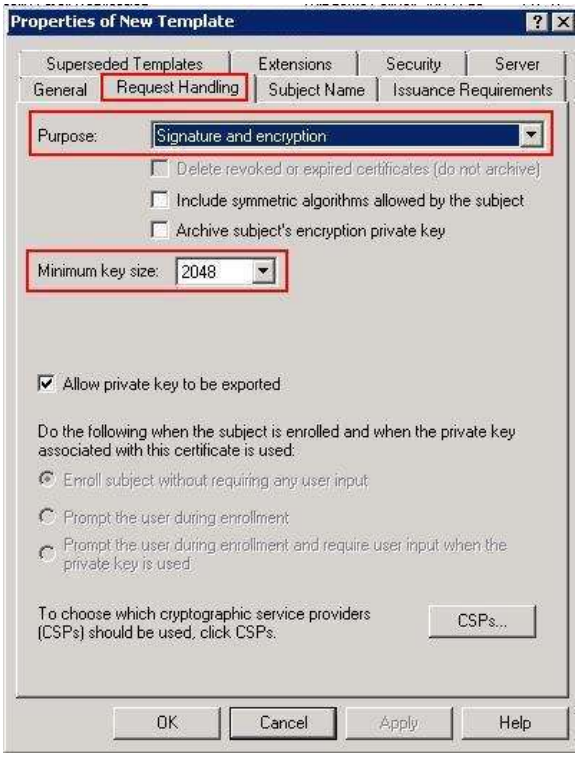


4. Select **Windows Server 2003 Enterprise** as the **Minimum Supported CA** and click **OK** (Not Shown)

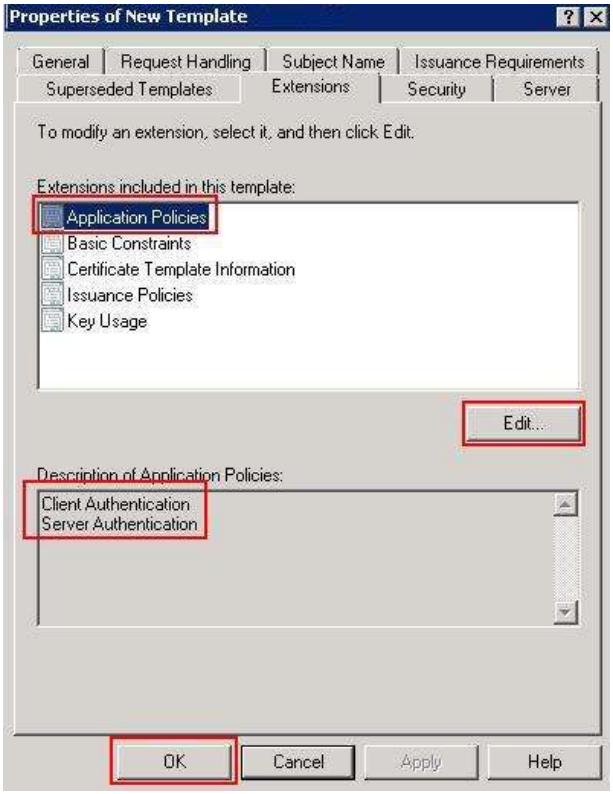
5. Give the template a name, example: **WebServer-Enterprise.** Select a **Validity Period** of **1 years** and select the checkbox beside **Publish Certificate in Active Directory**.



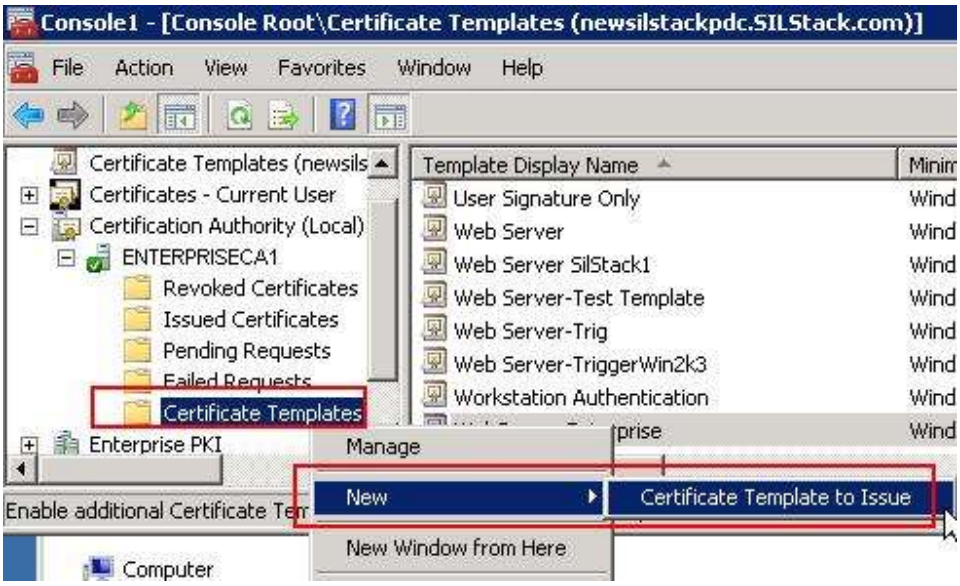
6. On the **Request Handling** tab, set the **Minimum key size** to **2048** and check the box beside **Allow private key to be exported**.



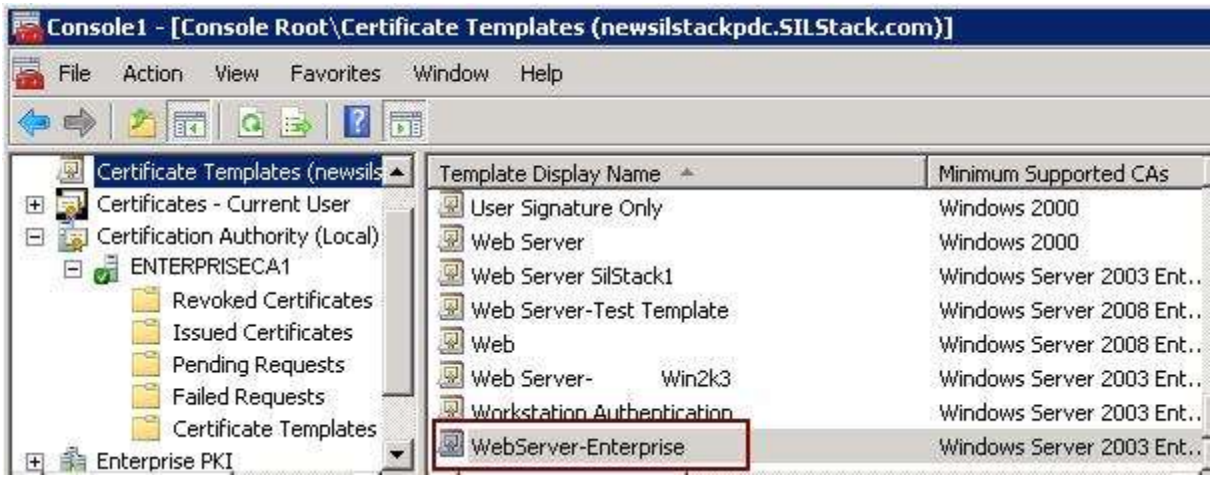
7. On the **Extensions** tab, click on **Application Policies** and click **Edit** to ensure both **Client Authentication** and **Server Authentication** are selected. Click **OK** to save the template.



On the console window, expand **Certification Authority (Local)**. Expand the server name and right-click on **Certificate Templates.** Select **New Certificate Template to Issue** to add the template created above.



The new template will now be shown in the list as shown below.



6. Configure Certificates for Avaya Aura® System Manager

This section describes the steps to replace default certificates on System Manager with certificates signed by a third-party CA. The steps involved are;

- Generate a Certificate Signing Request (CSR) and Private Key for System Manager
- Get the CSR signed by the CA
- Package the signed Identity Certificate and private key into a PKCS#12 archive format
- Install the third-party trusted Root certificate on System Manager
- Replace default System Manager Identity Certificate with the third-party signed identity certificate

6.1. Generate a Certificate Signing Request and Private Key for Avaya Aura® System Manager

In Public Key Infrastructure (PKI), a Certificate Signing Request (CSR) is a message sent to a CA containing certain information for a digital identity certificate. As part of the CSR process, a private key and public key (key pair) are created. The public key is sent as part of the CSR. The requirements for the Signed certificate for System Manager in this example are as follows;

- Naming Convention: x.509 PKI standards.
- Key Lengths: 2048 bit
- Hash Algorithms: X509 sha1 or sha256 (with RSA Encryption)
- CN = Fully Qualified Domain Name (FQDN)
- Subject Alternative Name = FQDN and vFQDN. This value for vFQDN or Virtual Fully Qualified Domain Name can be found in the following location on SMGR; **$MGMT_HOME/infra/conf/smgr-properties.properties**

6.1.1. Edit the OpenSSL Default Configuration File

OpenSSL is an open source program built into System Manager and it provides a utility to manage basic cryptographic functions. It's not possible to input the Subject Alt Name (SAN) using the basic openSSL interactive prompt as SAN is part of openSSLversion 3. Instead, it's necessary to use a configuration file. Avaya recommends entering two DNS Subject Alternative names. The first is the Fully Qualified Domain Name (FQDN) of System Manager and the second is used for Geographic Redundancy. The Geo Redundant name is normally the FQDN of System Manager preceded by the letters "gr" and is referred to as the virtual FQDN or vFQDN. A Geo Redundant name must be configured, even if System Manager is not configured for Geo Redundancy. In this case, it is not necessary to add the vFQDN to the DNS server.

Connect to System Manager using Secure Shell (SSH) for command line (CLI) access.

1. Log into System Manager using SSH connection as **admin**
2. Switch user to **root**
3. Make a backup copy of the default **openssl** configuration file located in following directory: **etc/pki/tls/openssl.cnf**
4. Edit the **openssl** configuration file.

An example of the configuration is shown as follows with the highlighted items in bold text showing the edits to the default configuration file;

```
login as: admin
Password:
[admin@smgr ~]$ su root
Using major release number R016x on System Platform
Password:
[root@smgr admin]# cp /etc/pki/tls/openssl.cnf /etc/pki/tls/openssl_default.cnf
[root@smgr admin]# vi /etc/pki/tls/openssl.cnf
 ……..

####################################################################
[ req ]
default_bits            = 2048
default_md              = sha1
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix   : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
# we use PrintableString+UTF8String mask so if pure ASCII texts are used
# the resulting certificates are compatible with Netscape
string_mask = MASK:0x2002

req_extensions = v3_req # The extensions to add to a certificate request
        ……..

[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment, dataEncipherment,
keyAgreement
extendedKeyUsage=serverAuth, clientAuth
subjectAltName= @alt_names

[alt_names]
DNS.1 = smgr.silstack.com
DNS.2 = grsmgr.silstack.com
```

6.1.2. Generate the CSR and Private Key for System Manager

On System Manager CLI, run the following command to generate the csr and a private key;
cd /home/admin
[root@smgr admin]# **openssl req -out SMGR.csr -new -newkey rsa:2048 -nodes -keyout SMGR.key -config /etc/pki/tls/openssl.cnf**
where

- **SMGR.csr** is the name of the CSR file output.
- **SMGR.key** is the name of the private key file.

This command requests input for various parameters such as C=country code, O=organisation, OU=Organisation Unit, etc. Enter the FQDN of System Manager as the Common Name (CN). Example CN = **smgr.silstack.com**.
The user is prompted to enter a challenge password for the private key.
The following is an example output;

> [root@smgr admin]# openssl req -out SMGR.csr -new -newkey rsa:2048 -nodes -keyout SMGR.key -config /etc/pki/tls/openssl.cnf
> Generating a 2048 bit RSA private key
> .....+++
> .....+++
> writing new private key to 'SMGR.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) [GB]:**US**
> State or Province Name (full name) [Berkshire]:**Colorado**
> Locality Name (eg, city) [Newbury]:**Westminster**
> Organization Name (eg, company) [My Company Ltd]:**Avaya**
> Organizational Unit Name (eg, section) []:**SIL**
> Common Name (eg, your name or your server's hostname) []:**smgr.silstack.com**
> Email Address []:
>
> Please enter the following 'extra' attributes
> to be sent with your certificate request
> A challenge password []:**xxxxxx**
> An optional company name []:

Verify the resulting CSR file. Issue the following command to print out the CSR file contents to screen and verify the details are correct;

openssl req –text –noout –verify –in SMGR.csr

Issue the following shell command to print the CSR file contents to the terminal window.
[craft@asm1 ~]$**cat SMGR.csr**

The output will be similar to the following;

-----BEGIN CERTIFICATE REQUEST-----
MIIDNDCCAhwCAQAwazELMAkGA1UEBhMCSUUxETAPBgNVBAgTCENvbm5hY2h0MQ8w
DQYDVQQHEwZHYWx3YXkxDjAMBgNVBAoTBUF2YXlhMQwwCgYDVQQLEwNTSUwxGjAY
BgNVBAMTEWFzbTEuc2lsc3RhY2suY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAv/iM1or94I5vDonMcL6OTUgT7z9hiL2Nya9KjNjbynOXE1jhfEsq
N69Gr6JGvtsF4r4p/lH4jlAZ9N1TNRuCcNmXAYBx9UAl9moj4EO93WC1nKcxkn2B
4BqUb5OdOQ8ImMqyDGp3jbCHxb5GiM4zUav34cOf6caPv+iBvf4hK51FnMUklJSY
JFs0+SwKYxS2b+nPolMPnLzYmEAXtVKukF0ogbJgLfYe0K18NC1OPdWJHXf0K0bX
5mmE3wPv0WehCIUp4HBbQzvnsybH8lR0sNqUo7sFCeoXixwuYSBlUefdOC11xMJC
0iAKvBNXOEPfntPfFIfKJYwsXCKNXFuOwQIDAQABoIGDMBgGCSqGSIb3DQEJBzEL
DAlBdmF5YTEyMyQwZwYJKoZIhvcNAQkOMVowWDAMBgNVHRMEBTADAQH/MAsGA1Ud
DwQEAwID+DAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIwHAYDVR0RBBUw
E4IRYXNtMS5zaWxzdGFjay5jb20wDQYJKoZIhvcNAQEFBQADggEBACXHenAXiEER
nhYGr8r4bDvYPzFfXwhofkl56gfuV8L7VoUBI4C+86v4DX6F2JseuPE/NVW0xx0c
h0S4TCJUktIL5oWxe6FLxYwMHXbnVhO64IkZzZ9TVC38e9eisgXMTFyNGl1qAE+x
Qa4pmpvKJrjJIGz4cZiRaR1dL51Lwovmh4bQTEDnI/snw5IT/IDdovvTz+gXCMmH
L0bxMTpRQwwc3CalEqcG4ogtv1edfTxQI85hpbMuIbYzJQfaNX7SkolsmRC+O9bW
ACsaXpHPhpmsc6ecmSPPKbFOjIWdVzbSwdPBqX9QjMPWqk/rRd5s01ivMbQFd5nL
UZpc5IgI068=
-----END CERTIFICATE REQUEST-----

Copy all the text from **----BEGIN** up to and including **REQUEST----**
Use this copied text to paste into the certificate request in **Section 6.2**

Note: The default hash algorithm for signature generation in the openSSL configuration file is SHA-1 (160bit) and this is the algorithm tested as part of this application note. If SHA-2 256bit algorithm is desired, add the command line argument **–sha256** as shown in the example below.

[root@smgr admin]# **openssl req -out SMGR.csr -new -newkey rsa:2048 -nodes –sha256 -keyout SMGR.key -config /etc/pki/tls/openssl.cnf**

The CA Server will be configured to support SHA-256. If using Windows Server 2008 a new template should be created with **Request Hash: SHA256**.

6.2. Process Certificate Signing Request on Certificate Authority

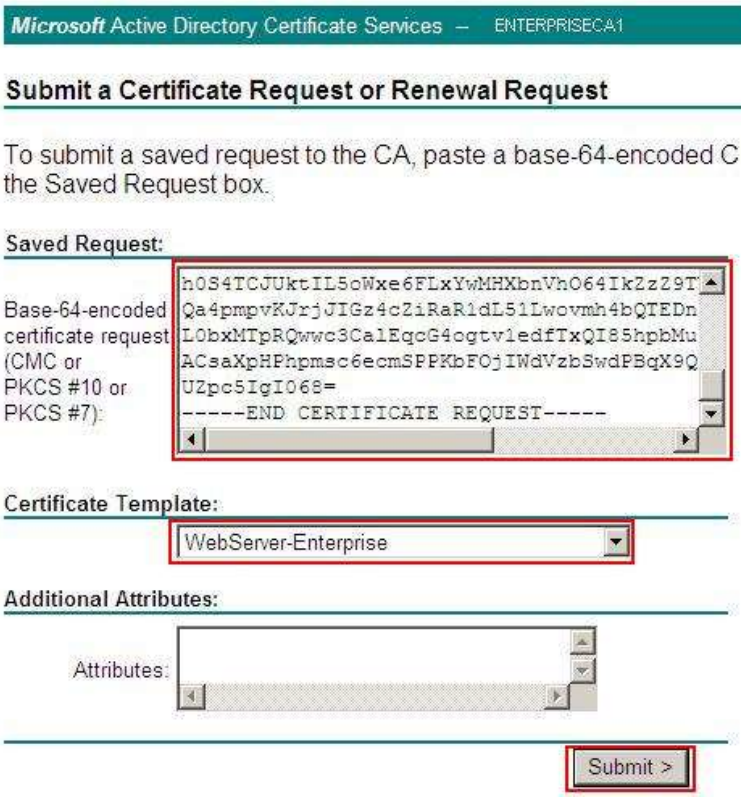
If a commercial third-party CA is used, the CSR file will be sent to this provider to be signed. Please contact the commercial CA service provider for details on this process. In this example, a CA within the Enterprise under the customers own control is being used.

Using Internet Explorer, browse to the Microsoft Active Directory Certificate Services on the CA server.

http://<IPaddressOfCAserver>/certsrv/
where **<IPaddressOfCAserver>** is the IP address or FQDN of the Microsoft Windows 2008 CA. Click on **Request a certificate**

Click on **Advanced Certificate Request** (Not Shown). Click 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**. (Not Shown)

Paste the contents of the CSR file **SMGR.csr** from **Section 6.1.2** into the **Base-64-encoded certificate** request box. Use the **WebServer-Enterprise** certificate template, created in **Section 5.2** and click **Submit**.



Select **Base64 encoded** radio button and click **Download certificate** to save file to the local PC. Save the file with **.pem** extension and a descriptive name, e.g., **SMGRsigned.pem**

While on the **CA Certificate Services** webpage, download the CA trusted Root file. From the default homepage, click on **Download a CA certificate, certificate chain, or CRL.**

Select the radio button beside **Base 64** encoding method and click on **Download CA Certificate**. Click **Save**, entering a descriptive name and a .pem extension. E.g., **CAroot.pem**, and save the file to the local PC.

6.3. Package the Private Key and Signed Certificate in a PKCS#12 format

System Manager expects a PKCS#12 format file when uploading a new identity certificate to the server. PKCS#12 is a password protected archive format used to store a number of cryptographic objects in a single file. Using an SFTP client, connect to System Manager and copy the signed identity certificate file **SMGRsigned.pem** and the trusted Root CA certificate **CAroot.pem** to System Manager home/admin directory. On System Manager CLI, use the **ls** command to ensure the private key file (SMGR.key), the signed identity certificate file (SMGRsigned.pem) and the

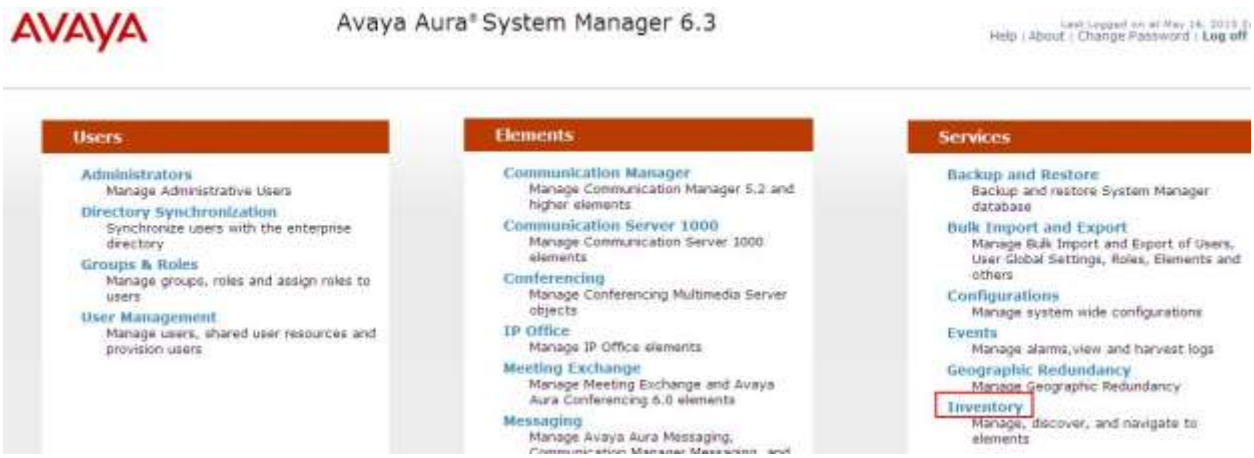
trusted Root CA certificate file (CAroot.pem) are present in the /home/admin directory. Issue the following command to create the PKCS#12 bundle;

openssl pkcs12 –export –out SMGR.p12 –inkey SMGR.key –in SMGRsigned.pem –certfile CAroot.pem

When prompted, enter a new password for the PKCS#12 archive file, to be used when importing this file into System Manager. Copy the resulting PKCS#12 format file SMGR.p12 to the local PC using an SFTP client or USB key.

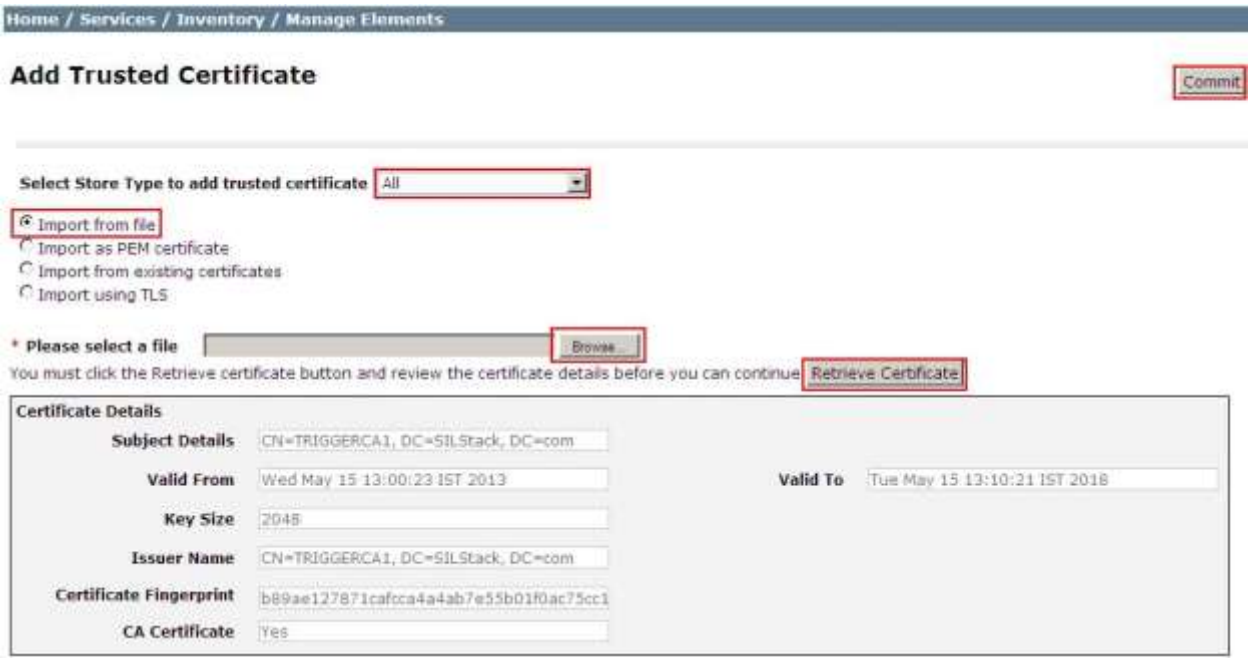
6.4. Install the trusted Root CA certificate in Avaya Aura® System Manager

Using a web browser, navigate to the System Manager web console by entering **https://<SMGRFQDN>**, where <SMGRFQDN> is the IP address or Fully Qualified domain name of System Manager. Enter the admin username and password. Under **Services,** click **Inventory.**



Click **Manage Elements** from the left navigation pane and select the checkbox beside **System Manager**. Click on the **More Actions** drop-down menu and select **Configure Trusted Certificates**.

Click **Add** (Not Shown). On **Add Trusted Certificate** page, select **All** for the **Select Store Type** drop-down menu. Select the radio button beside **Import from file**. Click **Browse** to locate the third-party CA root certificate file **CAroot.pem** on local PC and select **Retrieve Certificate** and then **Commit.** Click **Done** (Not Shown).



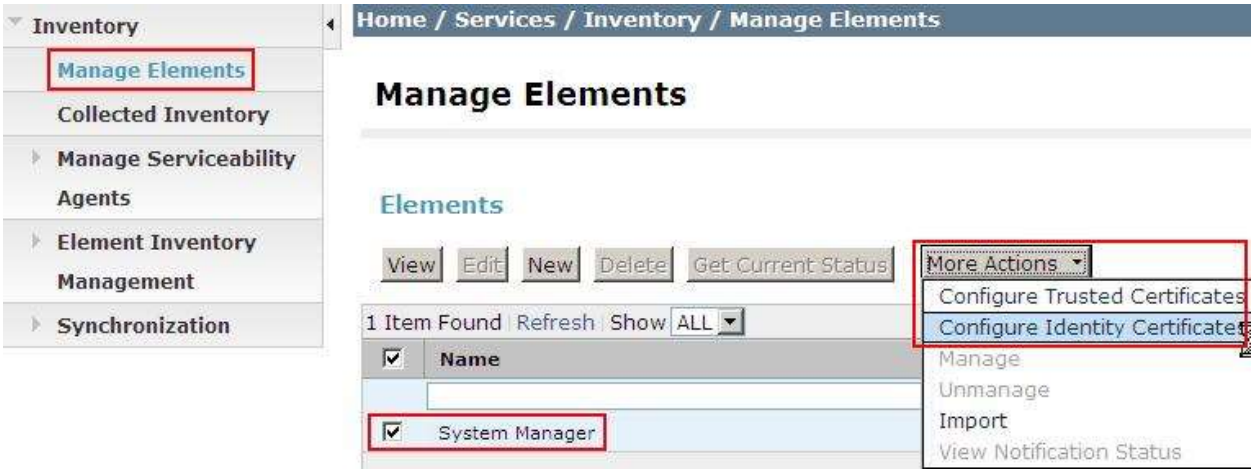
Access System Manager CLI via SSH, log in as admin and then switch user to root. Execute the following command;

#sh $SPIRIT_HOME/scripts/configureSpiritSecurity.sh

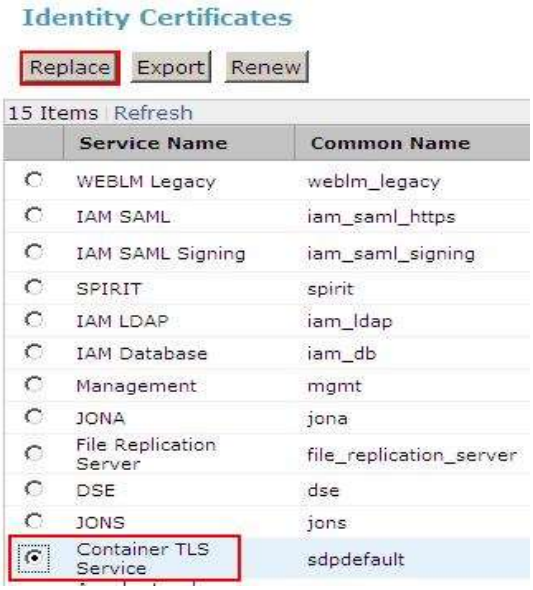
```
[root@smgr ~]# $SPIRIT_HOME/scripts/configureSpiritSecurity.sh
Stopping SPIRIT Agent Application 1.0-1.0...
Stopped SPIRIT Agent Application 1.0-1.0.
Starting SPIRIT Agent Application 1.0-1.0...
```

6.5. Replace Avaya Aura® System Manager Identity Certificate

Log into the System Manager web console. Under **Services**, click **Inventory.** Click **Manage Elements** and select the checkbox beside **System Manager.** Click **More Actions** -> **Configure Identity Certificates.**



Select the radio button beside **Container TLS Service** and click **Replace**.



Select **Import third party certificate** and beside **Please select a file (PKCS #12 format)** click **Browse**. Browse to the PKCS#12 file created in **Section 6.3**, **SMGR.p12** and enter the Password. Click **Retrieve Certificate**. Click **Commit**. Click **Done** (Not shown).

Depending on the customer security requirements, it may be necessary to replace the default identity certificate for other services, such as **Management**. Repeat these steps in **Section 6.5** to replace the default identity certificate with a third-party signed identity certificate for other services. Only **Container TLS Service** certificate replacement has been tested as part of this solution.

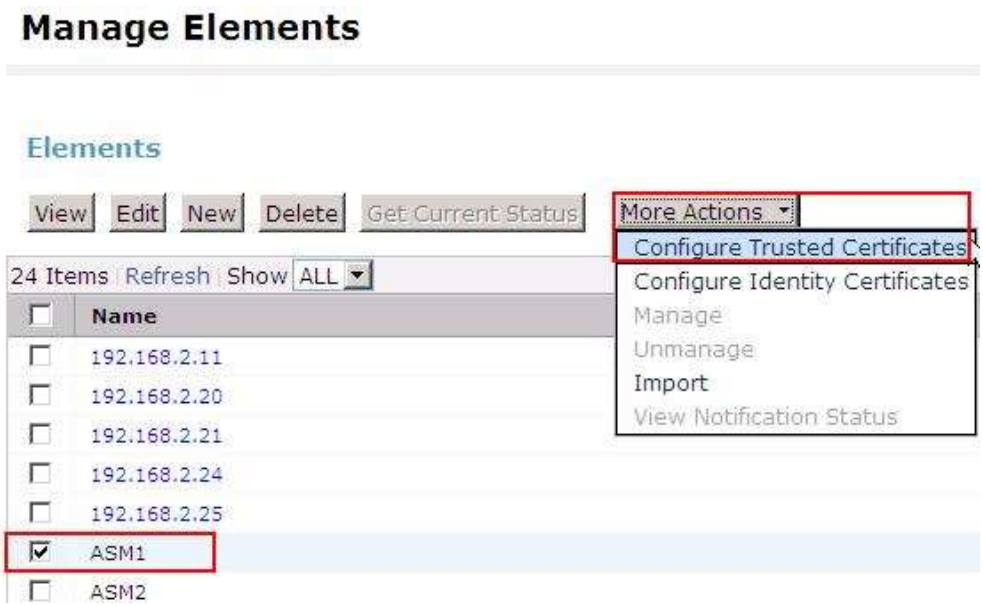
7. Configure Certificates for Avaya Aura® Session Manager

This section describes the steps to replace default certificates on Session Manager with certificates signed by a third-party CA. The steps involved are;

- Install the third party trusted Root certificate into Session Manager trusted store
- Restart System Manager JBoss Service
- Generate a Certificate Signing Request (CSR) and Private Key for Session Manager
- Get the CSR signed by the CA
- Package the signed Identity Certificate and private key into a PKCS#12 archive format
- Replace default Session Manager Identity Certificate with the third party signed identity certificate

7.1. Install third party Trusted Root CA Certificate on Avaya Aura® Session Manager

On the System Manager web console, under **Elements**, click **Inventory**. Click **Manage Elements** from the left navigation pane and select the checkbox beside first Session Manager, which is **ASM1** in sample network. Click **More Actions** and **Configure Trusted Certificates**.



On the trusted certificates page, click **Add.** On **Add Trusted Certificate** page, select **All** for the **Select Store Type** drop-down menu. Select the radio button beside **Import from File**. Click **Browse** to locate the third-party CA root certificate file **CAroot.pem** and select **Retrieve Certificate** and then **Commit**.

Click **Done** (Not shown). Access Session Manager CLI via SSH and log in as craft user. Switch user to sroot. Execute the following command to restart the Session Manager services;

#restart all

Repeat the procedure in this section for other Session Managers, such as ASM2.

7.2. Restart Avaya Aura® System Manager JBoss Service

At this stage of the process, both System Manager and Session Manager have the trusted root CA certificates installed. System Manager also has a third-party signed identity certificate. The JBoss webserver on System Manager can now be restarted and trust management between System Manager and Session Managers will use the third-party Certificate Authority. Log into System Manager CLI over SSH. Change to **root** user and issue the following command;

#service jboss restart

7.3. Generate a Certificate Signing Request and Private Key for Avaya Aura® Session Manager

In this section, the default **openSSL** configuration file will be edited and this configuration file will be used when issuing the command to generate a CSR and private key for Session Manager.

7.3.1. Edit the default OpenSSL Configuration File

As in **Section 6.1.1**, the default OpenSSL configuration file built into Session Manager will be edited to generate a key pair with 2048 bit key length. Connect to Session Manager using Secure Shell (SSH) for command line (CLI) access.

1. Log into Session Manager using SSH connection as **craft**
2. Switch user to **sroot**
3. Make a backup copy of the default **openssl** configuration file **etc/pki/tls/openssl.cnf**
4. Edit the **openssl** configuration file.

An example of the configuration is shown as follows with the highlighted items in bold text showing the edits to the default configuration file;

```
login as: craft
Password:
[craft@asm1 ~]$ su - sroot
Password:
[root@asm1 ~]# cp /etc/pki/tls/openssl.cnf /etc/pki/tls/openssl_default.cnf
[root@asm1 ~]# vi /etc/pki/tls/openssl.cnf
 ……..

##################################################################
[ req ]
default_bits            = 2048
default_md              = sha1
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix   : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
# we use PrintableString+UTF8String mask so if pure ASCII texts are used
# the resulting certificates are compatible with Netscape
string_mask = MASK:0x2002

req_extensions = v3_req # The extensions to add to a certificate request
        ……..

[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment, dataEncipherment,
keyAgreement
extendedKeyUsage=serverAuth, clientAuth
subjectAltName= @alt_names
```

```
[alt_names]
DNS.1 = asm1.silstack.com
```

7.3.2. Generate a CSR and Private Key

On Session Manager CLI, run the following command to generate the CSR;
[craft@asm1 ~]$ **cd /home/craft**
[craft@asm1 ~]$ **openssl req –out ASM1.csr –new –newkey rsa:2048 –nodes –keyout ASM1.key –config /etc/pki/tls/openssl.cnf**
where

- **ASM1.csr** is the name of the CSR file output.
- **ASM1.key** is the name of the private key file

Enter responses to the various parameters such as C=country code, O=organisation, OU=Organisation Unit, etc. Input the FQDN of Session Manager as the Common Name (CN).
Example CN = **asm1.silstack.com**
The user is prompted to enter a challenge password for the private key.
The following is an example output;

[craft@asm1 ~]$ openssl req -out ASM1.csr -new -newkey rsa:2048 -nodes -keyout ASM1.key -config /etc/pki/tls/openssl.cnf
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'ASM1.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) [GB]:**US**
State or Province Name (full name) [Berkshire]:**Colorado**
Locality Name (eg, city) [Newbury]:**Westminster**
Organization Name (eg, company) [My Company Ltd]:**Avaya**
Organizational Unit Name (eg, section) []:**SIL**
Common Name (eg, your name or your server's hostname) []:**asm1.silstack.com**
Email Address []:

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

Verify the resulting CSR file. Issue the following command to print out the CSR file contents to screen and verify the details are correct;
[craft@asm1 ~]$ **openssl req –text –noout –verify –in ASM1.csr**

Issue the following shell command to print the CSR file contents to the terminal window.
[craft@asm1 ~]$**cat ASM1.csr**

The output will be similar to the following;

-----BEGIN CERTIFICATE REQUEST-----
MIIDNDCCAhwCAQAwazELMAkGA1UEBhMCSUUxETAPBgNVBAgTCENvbm5hY2h0MQ8w
DQYDVQQHEwZHYWx3YXkxDjAMBgNVBAoTBUF2YXlhMQwwCgYDVQQLEwNTSUwxGjAY
BgNVBAMTEWFzbTEuc2lsc3RhY2suY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEAv/iM1or94I5vDonMcL6OTUgT7z9hiL2Nya9KjNjbynOXE1jhfEsq
N69Gr6JGvtsF4r4p/lH4jlAZ9N1TNRuCcNmXAYBx9UAl9moj4EO93WC1nKcxkn2B
4BqUb5OdOQ8ImMqyDGp3jbCHxb5GiM4zUav34cOf6caPv+iBvf4hK51FnMUklJSY
JFs0+SwKYxS2b+nPolMPnLzYmEAXtVKukF0ogbJgLfYe0K18NC1OPdWJHXf0K0bX
5mmE3wPv0WehCIUp4HBbQzvnsybH8lR0sNqUo7sFCeoXixwuYSBlUefdOC11xMJC
0iAKvBNXOEPfntPfFIfKJYwsXCKNXFuOwQIDAQABoIGDMBgGCSqGSIb3DQEJBzEL
DAlBdmF5YTEyMyQwZwYJKoZIhvcNAQkOMVowWDaMBgNVHRMEBTADAQH/MAsGA1Ud
DwQEAwID+DAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUHAwIwHAYDVR0RBBUw
E4IRYXNtMS5zaWxzdGFjay5jb20wDQYJKoZIhvcNAQEFBQADggEBACXHenAXiEER
nhYGr8r4bDvYPzFfXwhofkl56gfuV8L6VoUBI4C+86v4DX6F2JseuPE/NVW0xx0c
h0S4TCJUktIL5oWxe6FLxYwMHXbnVhO64IkZzZ9TVC38e9eisgXMTFyNGl1qAE+x
Qa4pmpvKJrjJIGz4cZiRaR1dL51Lwovmh4bQTEDnI/snw5IT/IDdovvTz+gXCMmH
L0bxMTpRQwwc3CalEqcG4ogtv1edfTxQI85hpHMuIbYzJQfaNX7SkolsmRC+O9bW
ACsaXpHPhpmsc6ecmSPPKbFOjIWdVzbSwdPBqX9QjMPWqk/rRd5s01ivMbQFd5nL
UZpc5IgI068=
-----END CERTIFICATE REQUEST-----

Copy all the text from **----BEGIN** up to and including **REQUEST----**
Use this copied text to paste into the certificate request in **Section 7.4.**

Note: The default hash algorithm for signature generation in the openSSL configuration file is SHA-1 (160bit) and this is the algorithm tested as part of this application note. If SHA-2 256bit algorithm is desired, add the command line argument **–sha256** as shown in the example below.

[craft@asm1 ~]$ **openssl req –out ASM1.csr -new -newkey rsa:2048 -nodes –sha256 -keyout ASM1.key -config /etc/pki/tls/openssl.cnf**

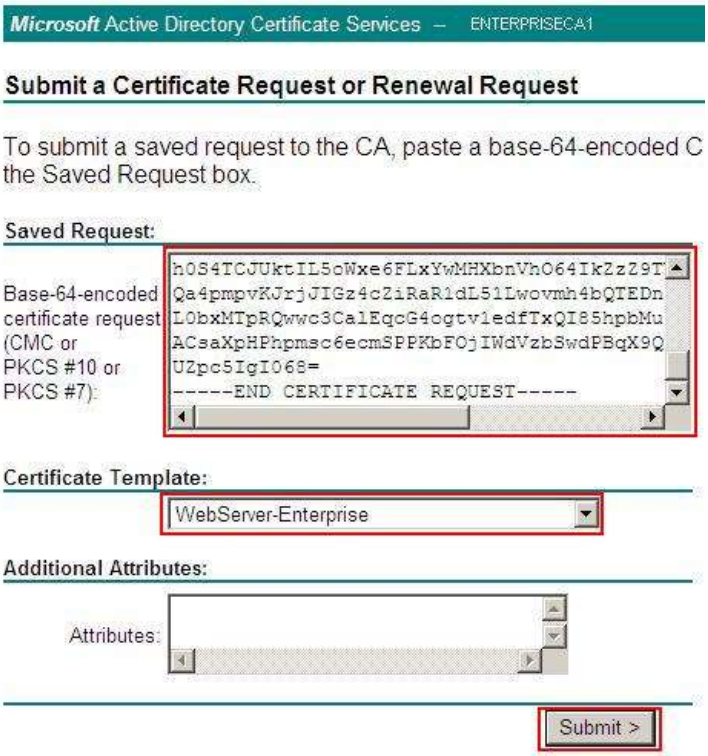
The CA Server will be configured to support SHA-256. If using Windows Server 2008 a new template should be created with **Request Hash: SHA256**.

7.4. Process the Certificate Signing Request on Certificate Authority

Using Internet Explorer, browse to the Microsoft Active Directory Certificate Services on the CA server using following url: **http://<IPaddressOfCAserver>/certsrv/**
Click on **Request a certificate**.

Click on **Advanced Certificate Request** (Not shown). Click 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**. (Not Shown).

Paste the contents of the CSR file **ASM1.csr** file from **Section 7.3.2** into the **Base-64-encoded certificate** request box. Use the **WebServer-Enterprise** certificate template, created in **Section 5.2** and select **Submit**.



Select **Base64 encoded** radio button and click **Download certificate** to save file to the local PC (Not Shown). Save the file with **.pem** extension and a descriptive name, e.g., **ASM1signed.pem**.

7.5. Package the Session Manager Private key and Signed Certificate in PKCS#12 format

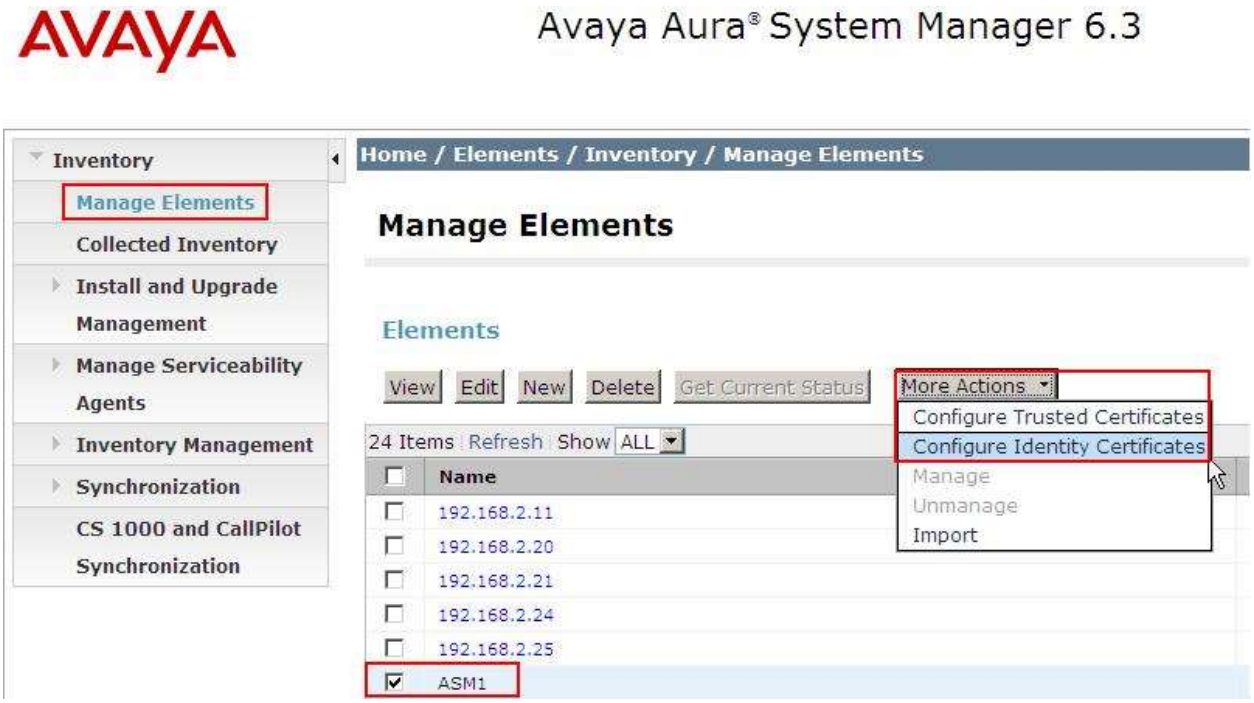
Use an SFTP client or USB key to copy this signed certificate file **ASM1signed.pem** from the local PC to the home folder /home/craft on Session Manager. On Session Manager CLI, issue the following command to create the PKCS#12 bundle;

openssl pkcs12 –export –out ASM1.p12 –inkey ASM1.key –in ASM1signed.pem

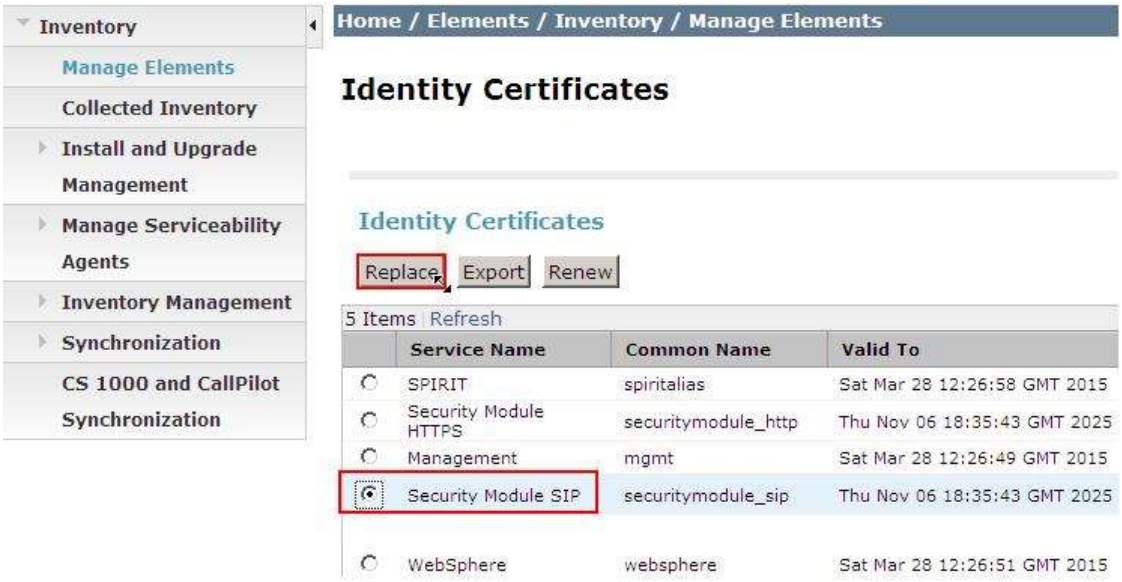
When prompted, enter a password to export this archive file. Using an SFTP client or USB key, copy this PKCS#12 format file ASM1.p12 to the local PC.
Repeat this procedure for second Session Manager.

7.6. Replace the Default Avaya Aura® Session Manager Identity Certificate

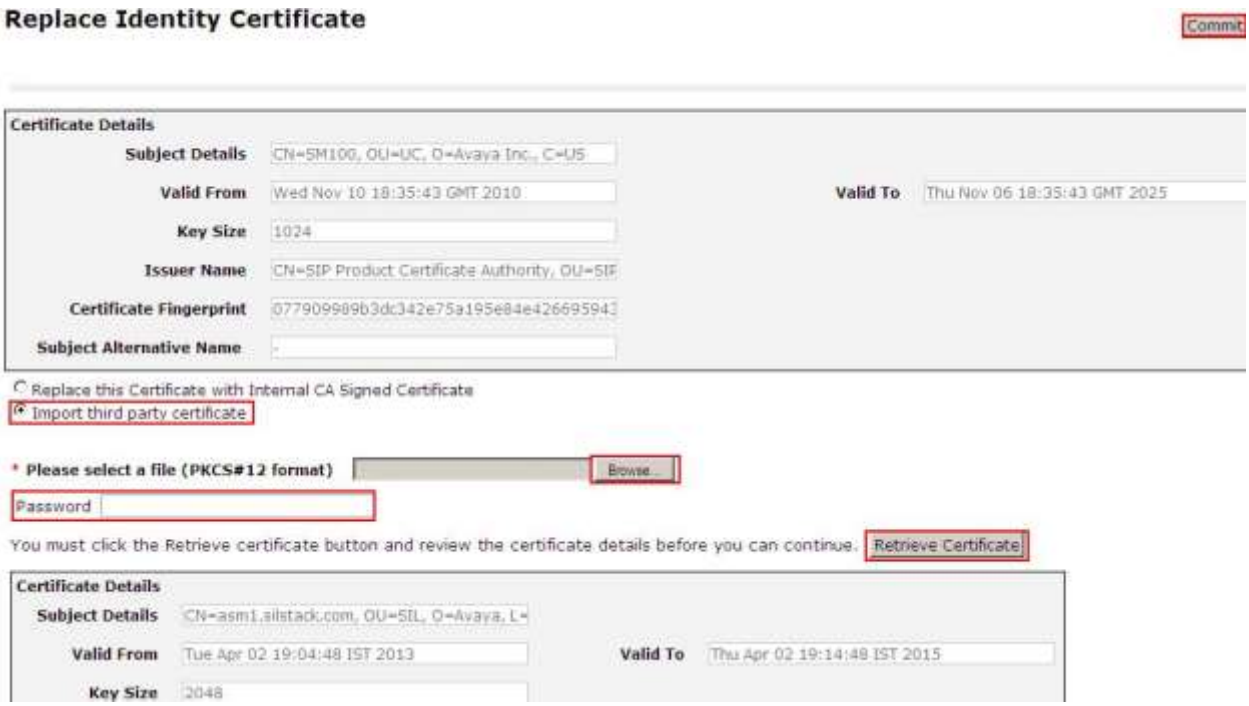
Session Manager contains a default Identity certificate with a hardcoded Common Name (CN) **sm100** for SIP communication. Each Session Manager will need to be changed to use a third-party signed identity certificate with its unique FQDN as the Common Name on the certificate. On System Manager web console, navigate to **Inventory → Manage Elements**. Select the check box beside the Session Manager element ASM1. Select **Configure Identity Certificates** from **More Actions** menu as shown below.



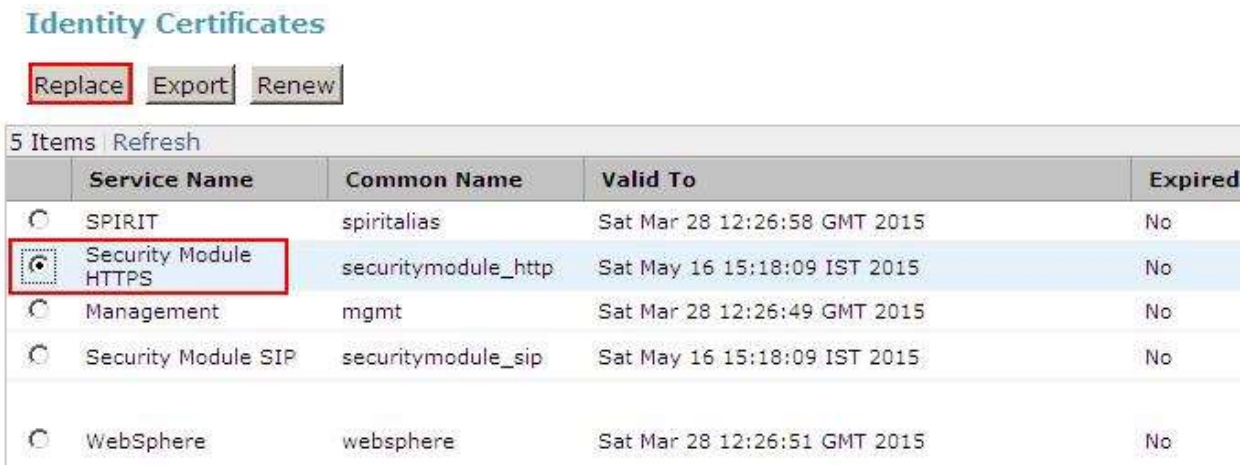
Select the radio button beside **Security Module SIP**. The details of the default Session Manager Security certificate are shown. Note **SM100** as the CN. Click on the **Replace** button in order to replace this default identity certificate with a customer defined or third-party certificate.



Select **Import third party certificate**. Click **Browse** to locate the PKCS#12 file created in **Section 7.5** (i.e. ASM1.p12), enter the key import password and click **Retrieve Certificate**. Click on **Commit** and **Done** on the following screen (Not shown).



PPM data exchange with Session Manager occurs over HTTPS, port 443. TLS certificate exchange for PPM should also use the third-party certificates. Select the radio button beside S**ecurity Module HTTPS** as shown below. The details of the default Session Manager Security certificate are shown. Note **SM100** as the CN. Click on the **Replace** button in order to replace this default identity certificate with a customer defined certificate.



Select **Import third party certificate**. Click **Browse** to locate the same PKCS#12 file created in **Section 7.5** ASM1.p12, enter the key import password and click **Retrieve**. Click on **Commit** and **Done** on the following screen (Not shown).

Depending on a customer's security requirements, it may be necessary to configure unique identity certificates for both SIP and HTTP. In this example, the same third-party signed certificate was used for both SIP and HTTP.

8. Verification Steps

This section describes steps to confirm TLS is being used by System Manager and Session Manager and to verify the correct third-party signed certificates are in use.

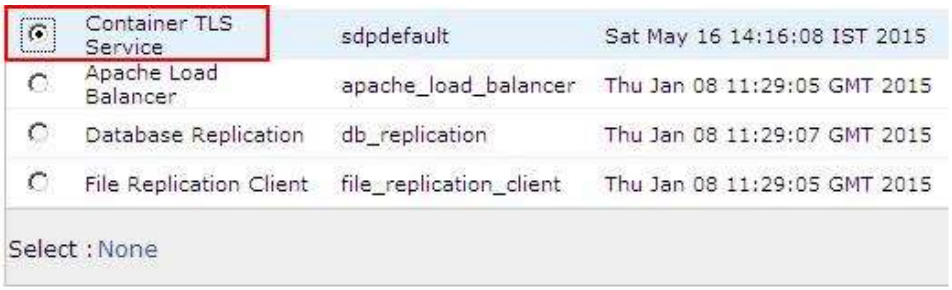
8.1. Avaya Aura® System Manager Verification

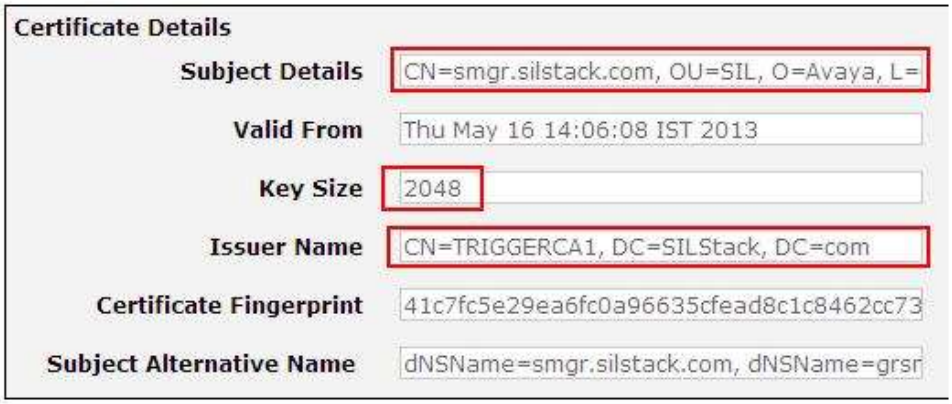
The steps to verify System Manager is communicating using TLS with third-party certificates are;

- Verify third-party identity certificate is in use.
- Verify third-party/customer CA is in the trusted certificate store.
- Verify certificate exchange with Session Manager using packet capture.

Log into System Manager web console and select **Services→Inventory→Manage Elements**. Select the check box beside the System Manager. Click on **More Options** and select **Configure Identity Certificate.**

Select **Container TLS Service** and check the details of the certificate are correct. The **Subject Details** should match the information provided to the third-party or customer CA as part of the CSR. The key length should be 2048 bit, if this was requested as part of the CSR. The **Issuer Name** should contain the name of the third-party or customer CA and domain information.





Within System Manager web console, select **Services→Inventory→Manage Elements**. Select the check box beside the **System Manager**. Click on **More Options** and select **Configure Trusted Certificates**.

Verify a trusted certificate exists for System Manager Trust Management with the **Subject Name** of the third-party or customer CA for both **TM_INBOUND_TLS**, **TM_OUTBOUND_TLS,** and **TM_INBOUND_TLS_PEM**.

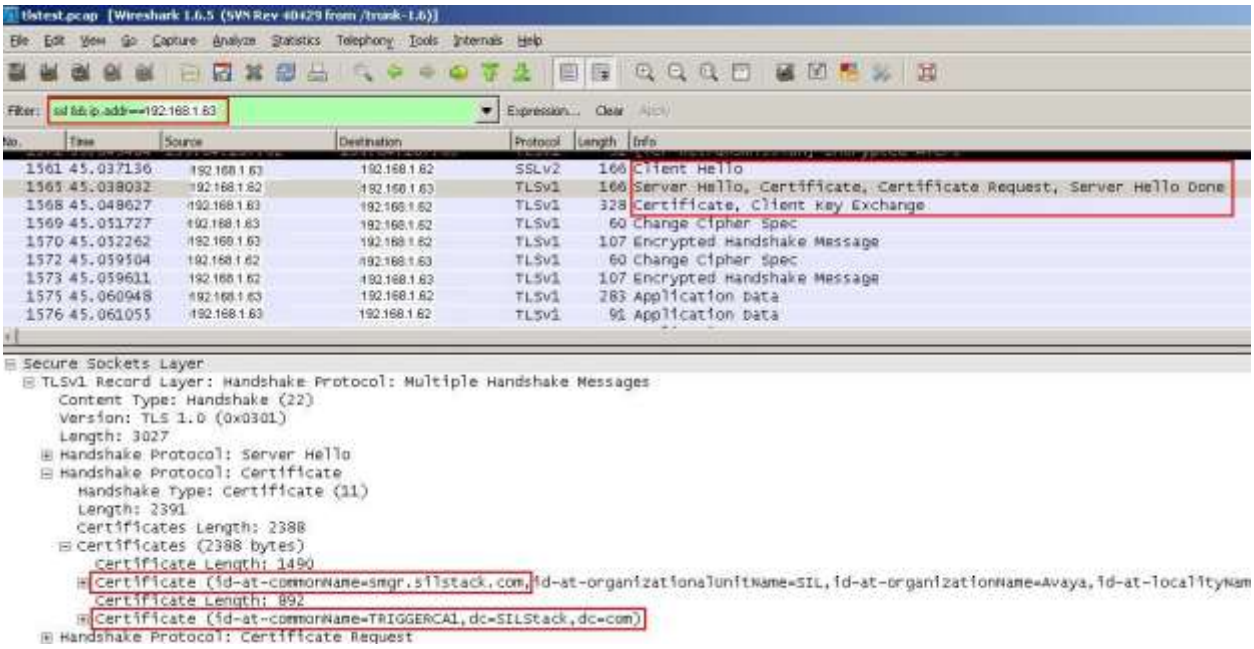
Log into System Manager CLI using Secure Shell (SSH) application, as admin user. Switch user to **root** and issue the following command to start a packet capture on System Manager Ethernet interface 0 and log the output to a file called **tlstest.pcap**

[root@smgr ~]# /usr/sbin/tethereal -i eth0 -w /home/admin/tlstest.pcap

Leave the packet capture running for five minutes. It may be necessary to restart Session Manager Security module to force a TLS handshake. See **Section 10 [2]** for details on how to restart Session Manager Security Module. Ensure administrator has approved a restart of Security Module.

Stop the capture by issuing **Ctrl+C** (Press “Ctrl” keyboard button and “c” button at the same time). Issue the command **chown admin /home/admin/tlstest.pcap** to change the packet capture file ownership from root to admin user. Using an SFTP client application (e.g., Filezilla or WinSCP), copy the packet capture file from /home/admin to the local PC.

Use a packet capture analysis tool, such a Wireshark, to open the capture and graphically view the packets. Filter for **ssl** and the management IP address of Session Manager. Search for a **Server Hello** sent from System Manager to Session Manager. In the details pane for this packet, expand **Secure Sockets Layer** and **Certificates**. This shows the certificates offered by System Manager as part of TLS three-way handshake. As shown on the screenshot below, the common name of the System Manager Identity certificate, signed by the third-party CA and the third-party CA trusted root certificate are offered.

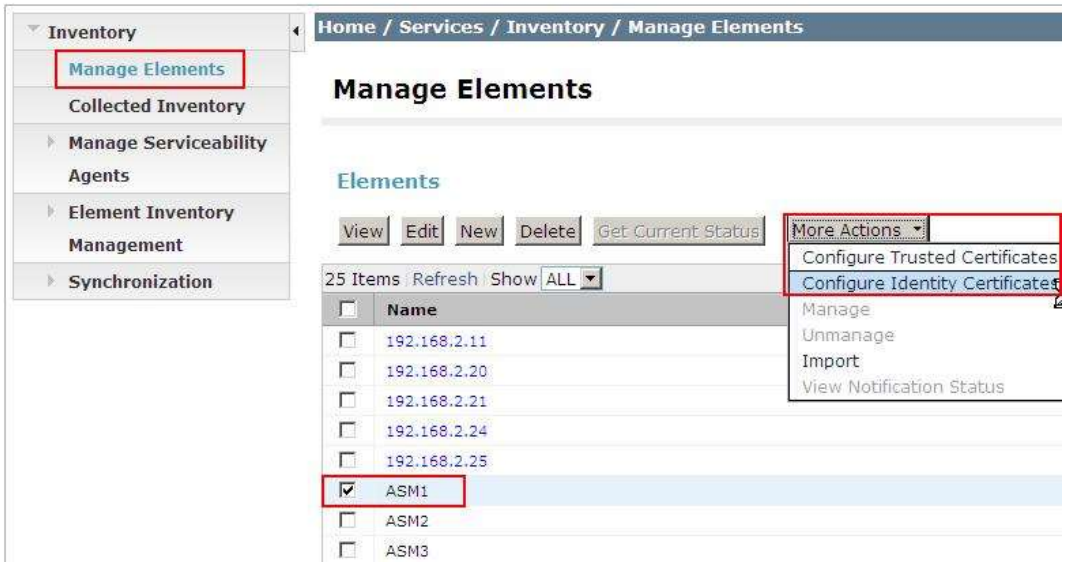


8.2. Avaya Aura® Session Manager Verification

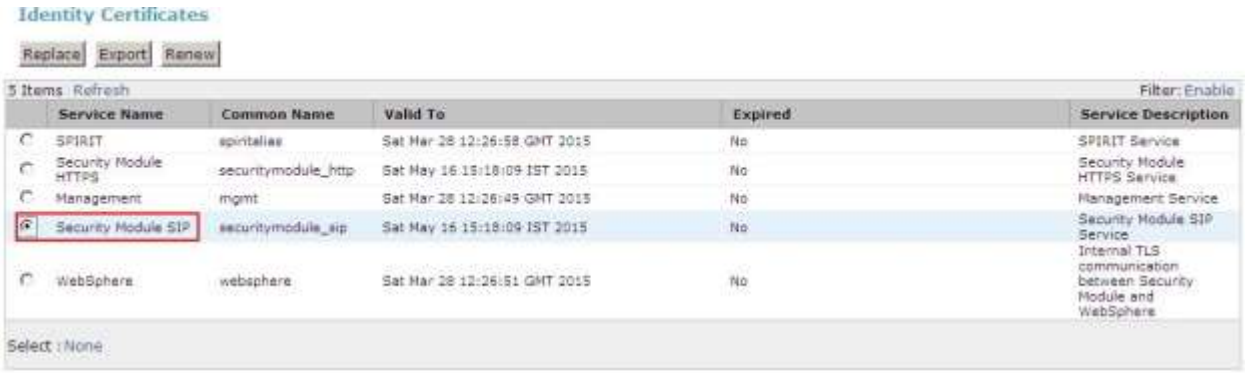
The steps to verify Session Manager is communicating using TLS with third-party certificates are;

- Verify third-party identity certificate is in use.
- Verify third-party/customer CA is in the trusted certificate store
- Verify Session Manager Security Module is using a Customer CA
- Using a packet capture tool, verify the correct certificates are used and TLS handshake is successful

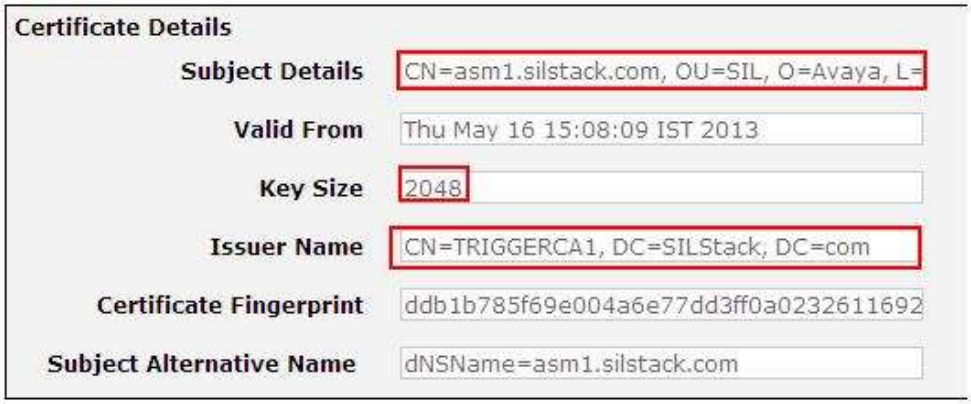
Log into System Manager web console. Select **Services→Inventory→Manage Elements**. Select the check box beside the Session Manager. Click on **More Options** and select **Configure Identity Certificate**



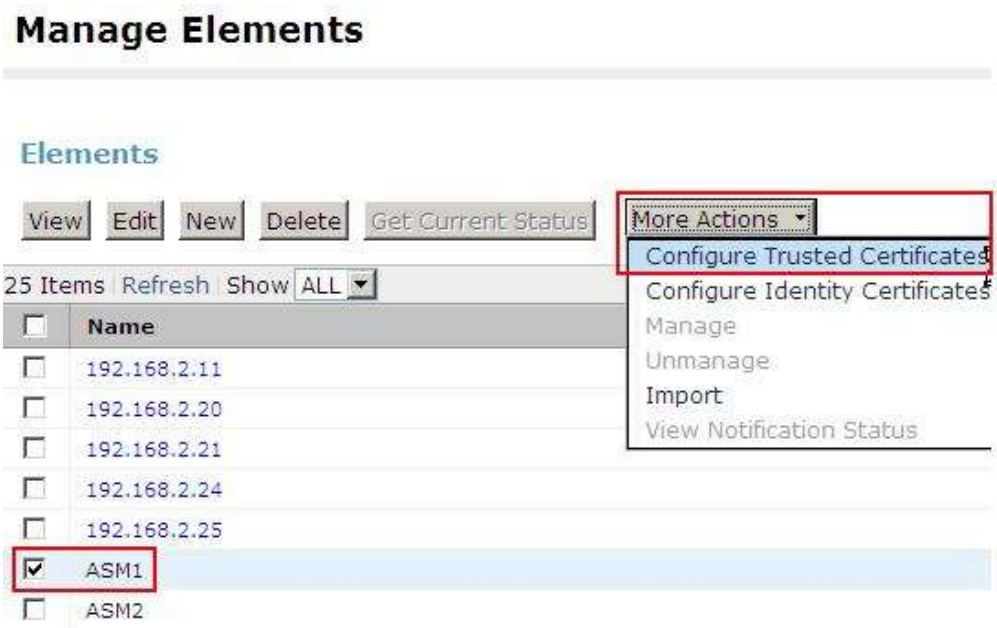
Select the **Security Module SIP**. The certificate details are displayed. Ensure the details of the certificate are correct as provided by the third-party or customer Certificate Authority. Verify the certificate date is valid, the **Subject Details** match the request sent to the third-party CA, the **Key Size** is the correct length (as per the CSR) and the **Issuer Name** matches that of the third-party CA.



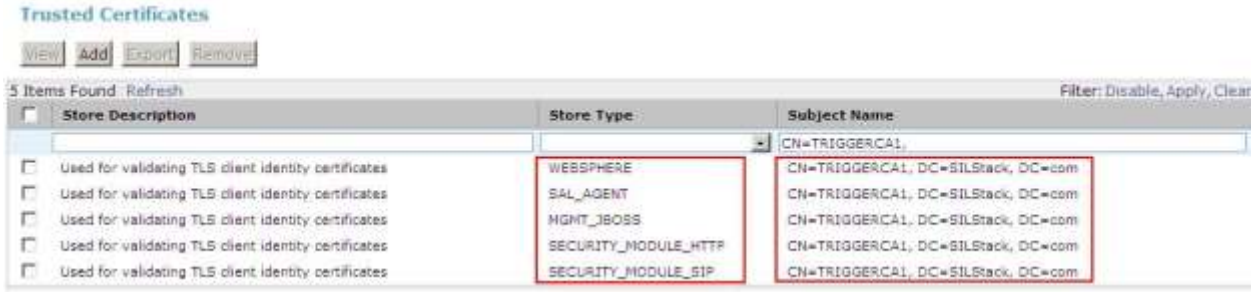
Select the **Security Module HTTPS** and verify the certificate details are correct.



Within System Manager web console, select **Services→Inventory→Manage Elements.** Select the check box beside the Session Manager **ASM1**. Click on **More Options** and select **Configure Trusted Certificates**.



Ensure an entry exists in each **Store Type** for the third-party or customer CA trusted root certificate.

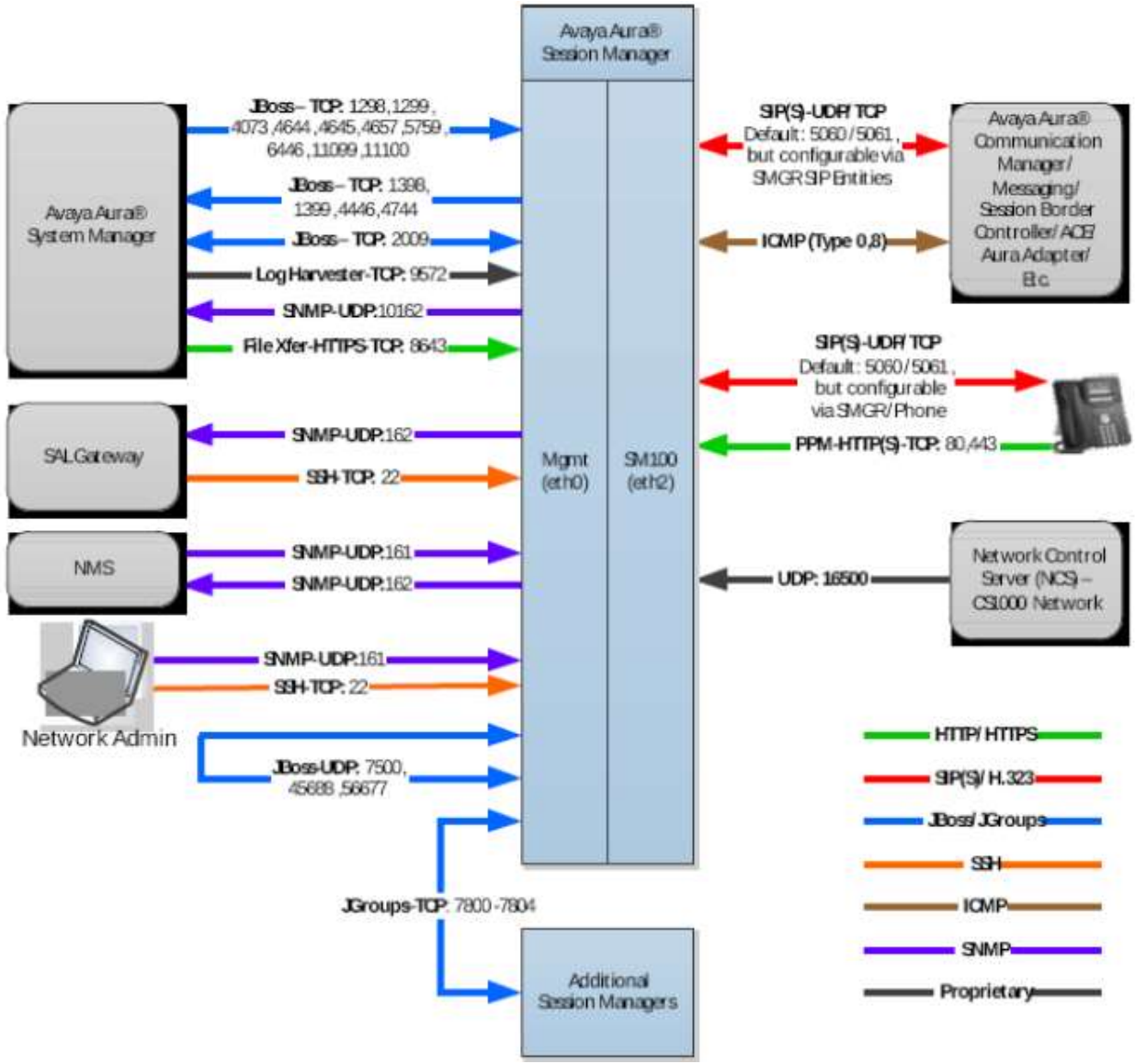


Verify Session Manager is using the **Customer CA** rather than the default certificate issued by Avaya SIP CA. On System Manager web console, navigate to **Elements→ Session Manager→ System Status→Security Module Status**. Ensure the **Certificate Used** for **ASM1** is **Customer CA**, as shown below.

Using a Terminal Emulator such as PuTTY, log into Session Manager Secure Shell as craft user. Change user to **sroot**. Issue the following command;

/usr/sbin/tethereal -i eth2 port 5061 -w /home/craft/asmtlstest.pcap

Where **asmtlstest.pcap** is the name of the packet capture file. and **eth2** is the Session Manager Ethernet interface to capture packets on. Port **5061** is the port on interface 2 used for SIP-TLS. Session Manager communicates with System Manager over Management Ethernet interface 0. Session Manager SM100 Security Module communicates with SIP endpoints on Ethernet interface 2, as shown below.



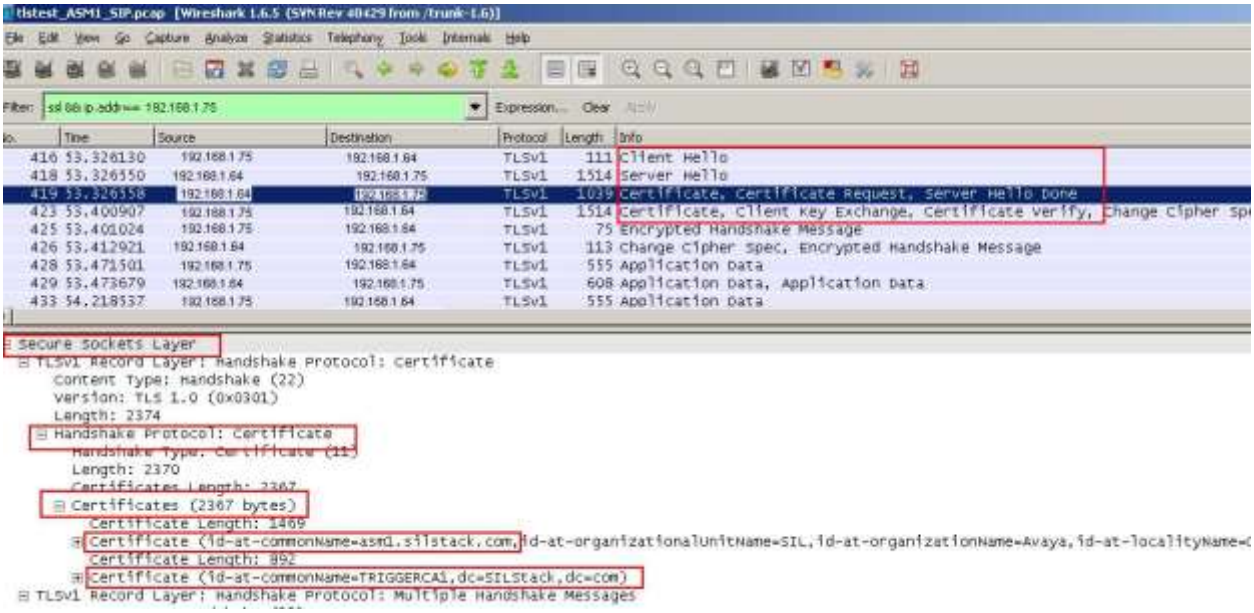
To observe TLS certificate exchange, it is necessary to capture packets at the start of the TLS handshake sequence, or else observe a session refresh. Since session refreshes are asynchronous,

either a client or a server service restart is required to guarantee capturing a TLS handshake. To enforce as TLS handshake for test purposes, restart the far end SIP application connecting to Session Manager. An example is Avaya Aura® Communication Manager.

NOTE: Confirm with systems administrator that it is acceptable to restart SIP application prior to beginning this procedure.

After two minutes stop the packet capture by issuing **Ctrl+C** (Press "Ctrl" keyboard button and "c" button). Issue the following command to change ownership of the packet capture file from root to craft user.
chown craft /home/craft/asmtlstest.pcap.

Copy the packet capture file from /home/craft to the local PC and open using Wireshark packet capture analysis application. Filter the capture for **ssl** and the IP address of the far end application/server connecting to Session Manager using SIP-TLS. Click on the packet sent from Session Manager as part of the Server Certificate offer and request as shown below. Note the Common Name of the certificate, signed by third-party CA and the trusted Root certificate of the third-party CA are offered by Session Manager.



9. Conclusion

These Application Notes describe how to configure Avaya Aura® System Manager 6.2 FP2 and Avaya Aura® Session Manager 6.2 FP2 to use TLS security certificates signed by a customer or third-party Certificate Authority. Microsoft Windows 2008 R2 Enterprise is configured as a Certificate Authority and examples are provided to illustrate the process of signing a certificate signing request (CSR) generated from Avaya Aura® System Manager and Avaya Aura® Session Manager.

10. Additional References

Avaya Product documentation relevant to these Application Notes is available at http://support.avaya.com.

[1] Administrating Avaya Aura System Manager, Release 6.3, Issue 2, may 2013

[2] Administering Avaya Aura Session Manager, Release 6.3 Issue 2, May 2013

[3] Avaya Aura® 6.2 Feature Pack 2 System Manager Release 6.3.2 Security Guide, Release 6.3.2, Issue 0.1, May, 2013

[4] Security Design in Avaya Aura® Session Manager, Release 6.3, October 2013

[5] Microsoft Technet on http://technet.microsoft.com

[6] RFC 5246 - The Transport Layer Security (TLS) Protocol
- available from http://www.ietf.org/

Please e-mail any questions or comments pertaining to these Application Notes along with the full title name and filename, located in the lower right corner, directly to the Avaya Solution & Interoperability Test Lab at interoplabnotes@list.avaya.com