Technical Report

NetApp Storage Encryption: Preinstallation Requirements and Procedures For IBM Tivoli Lifetime Key Manager (TLKMv2)

Mike Wong, NetApp
Neil Shah, NetApp
April 2013 | TR-3954

|Version 1.3|

NETAPP STORAGE ENCRYPTION PREINSTALLATION STEPS

NetApp® Storage Encryption (NSE) requires a number of components that must be completed prior to configuration within Data ONTAP®. This includes installation and configuration of a Key Management Interoperability Protocol key management server (KMIP server), SSL certificate creation and signing, and manual configuration of Data ONTAP bootloader variables. This guide offers a step-by-step example of the preinstallation steps using an IBM TKLM Server for Windows® as the KMIP server, OpenSSL for Linux® certificate generation, and Windows 2008 Certificate Authority for certificate signing.

1 INTRODUCTION

NetApp Storage Encryption has a number of preinstallation steps that must be completed before configuration in Data ONTAP can begin. These steps can be broken into the following main categories:

- Bootloader variable configuration in Data ONTAP
- SSL certificate creation
- SSL certificate signing
- Installation of signed SSL certificates to correct locations

Upon completion of these preinstallation steps, refer to the storage encryption section of the document "Data ONTAP 8.1 7-Mode Software Setup Guide" to complete setup of NSE.

2 REQUIRED BOOTLOADER VARIABLES FOR NSE

Data ONTAP has some specific commands that must to be run prior to running the setup wizard for NSE. Failure to configure these variables can result in loss of access to the encrypted disks until the values are added.

2.1 CONFIRM VERSION OF DATA ONTAP

NSE is compatible with 7-Mode Data ONTAP 8.1.x GA or greater and clustered Data ONTAP beginning with 8.2. Earlier versions of Data ONTAP will fail to recognize the disks in the system. The disk will show up in a FAILED state.

When running 7-Mode Data ONTAP 8.1.x GA or greater systems running NSE should not be downgraded below 8.1 or the disks will not be seen by the system.

When running clustered Data ONTAP 8.2, systems running NSE should not be downgraded to any prior versions of clustered Data ONTAP, or the disks will not be seen by the system.

2.2 CONFIGURE BOOTLOADER ENVIRONMENT VARIABLES

Data ONTAP requires certain boot environment variables to be configured prior to NSE setup.

BOOTARG.STORAGEENCRYPTION.SUPPORT

This bootarg is typically set during the manufacturing process. However, if the encrypted disks are not showing up at boot time, verify the preceding bootarg is set to true.

Halt Data ONTAP and stop at the LOADER-(A,B)> prompt.

Syntax to set the variable:

```
LOADER-A> setenv bootarg.storageencryption.support true
```

Example where variable is defined:

```
LOADER-A> printenv bootarg.storageencryption.support

Variable Name        Value
-------------------- ------------------------------------------------
bootarg.storageencryption.support true
```

Example where variable is not defined:

```
LOADER-A> printenv bootarg.storageencryption.support

Variable Name        Value
-------------------- ------------------------------------------------
bootarg.storageencryption.support *** Undefined ***
```

IP ADDRESS ENVIRONMENT VARIABLES

These bootargs need to be set so the FAS platform knows which Ethernet interface is used to communicate to the KMIP server for authentication key retrieval. This *is not* the IP address of the KMIP server.

These commands are also entered at the bootloader prompt.

Enter the following:

```
LOADER-A> setenv kmip.init.interface <interface>
LOADER-A> setenv kmip.init.ipaddr <IP Address of interface>
LOADER-A> setenv kmip.init.netmask <Netmask of interface>
LOADER-A> setenv kmip.init.gateway <Gateway of interface>
LOADER-A> saveenv
```

- **kmip.init.interface** is set to the Data ONTAP network interface you want to use. This interface must be dedicated for NSE use and cannot participate in network trunking or VIF configuration.
- **kmip.init.ipaddr** is set to the IP address of the interface in kmip.init.interface. Note that this will be the same IP address you assigned during Data ONTAP setup.
- **kmip.init.netmask** is the netmask for kmip.init.interface. This is the same netmask used in Data ONTAP setup.
- **kmip.init.gateway** is the gateway for kmip.init.interface. This is the same gateway used in Data ONTAP setup.

Once the bootloader variables have been configured, you are ready to start Data ONTAP. The subsequent sections provide guidance on creating SSL certificates to establish a secure communications channel between NSE and the key manager.

3 SSL CERTIFICATE CREATION

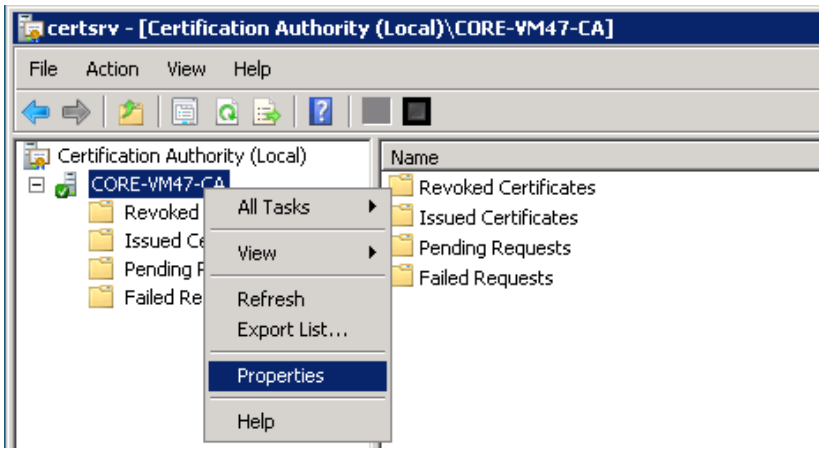
Secure Sockets Layer (SSL) certificates are used to establish trusted communications between parties. In this section, we will create the following SSL certificates, which will then need to be signed before use. This example uses a third-party certificate authority (CA) to sign the certificates. An example using self-signed certificates can be found in the appendix. The following SSL certificates will be generated:

- CA public certificate
  - This is an exported public certificate from the CA.
  - This file needs to be renamed <IP_Address_of_KMIP_Server>_CA.pem for NSE use.
- KMIP server public certificate
  - This needs to be generated at the KMIP server and usually results in generation of a private/public pair.
- NSE public certificate
  - This needs to be generated on any computer using OpenSSL in Windows or UNIX®.
  - This file needs to be renamed client.pem.
- NSE private certificate
  - This needs to be generated on any computer using OpenSSL in Windows or UNIX.
  - This file needs to be concatenated with the public certificate and renamed client_private.pem.

3.1 EXPORTING THE CA CERTIFICATE

This certificate is the public certificate of the certificate authority. It is needed by both NSE and the KMIP server to validate the signed certificates being exchanged. The following example shows how the CA certificate is obtained from a Windows 2008 CA server.

Figure 1-2: Exporting A Windows 2008 CA Certificate



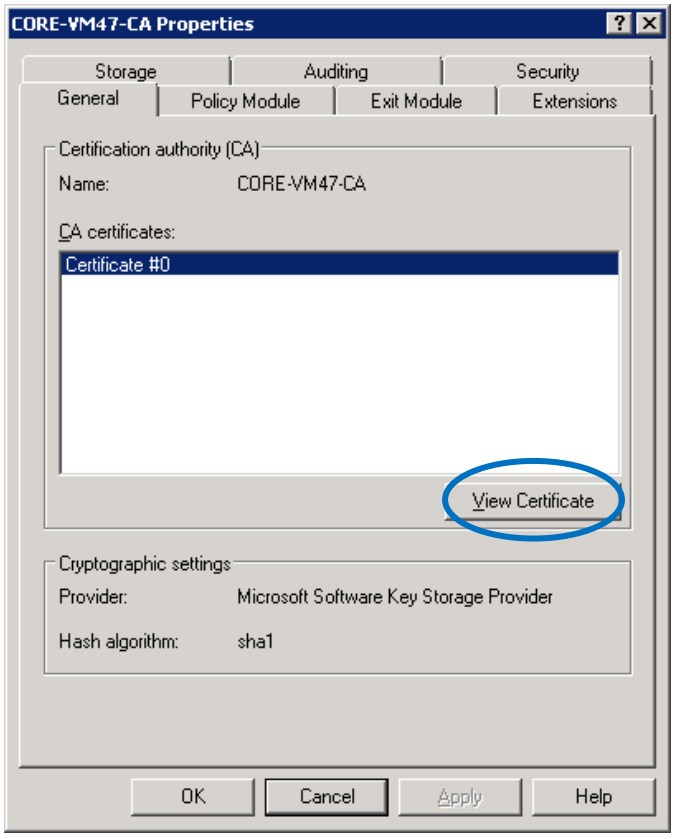
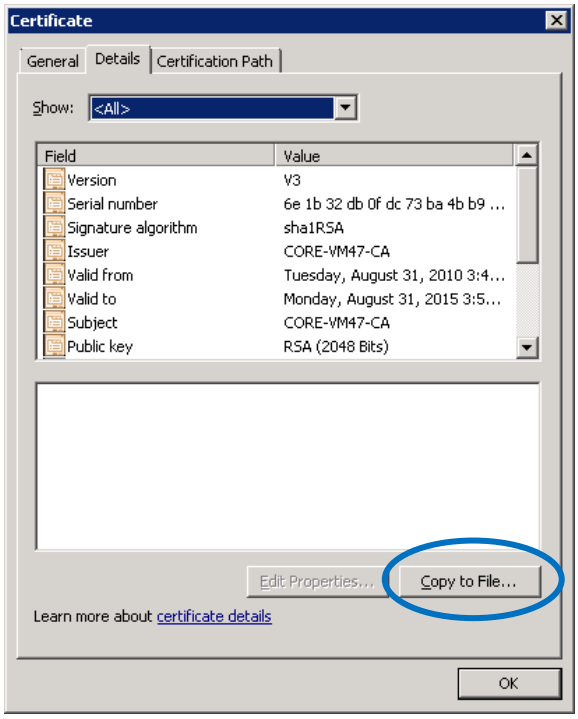
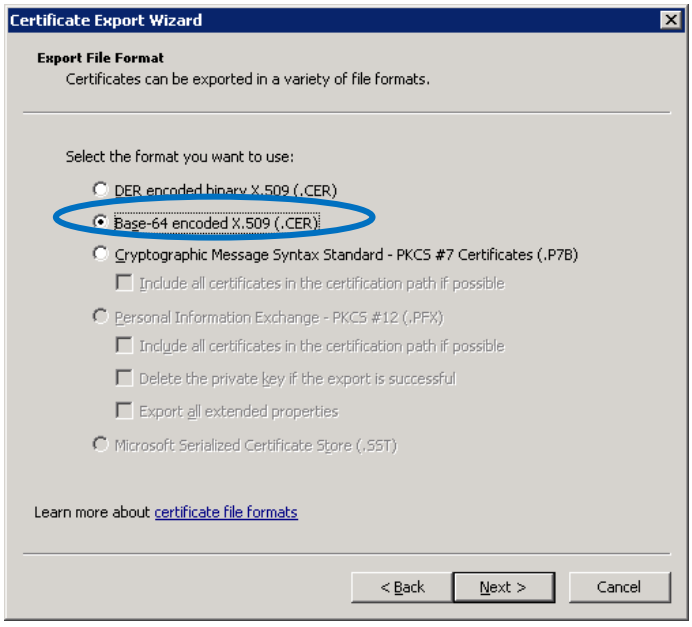


Figure 3-5: Certificate Export Wizard

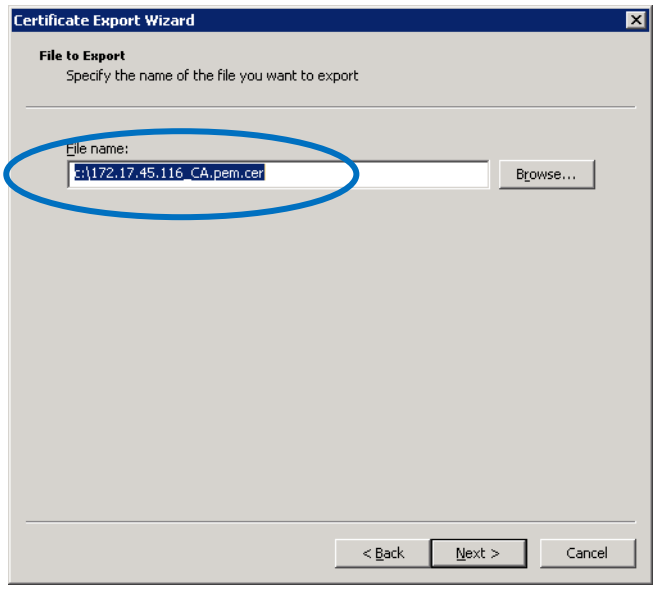


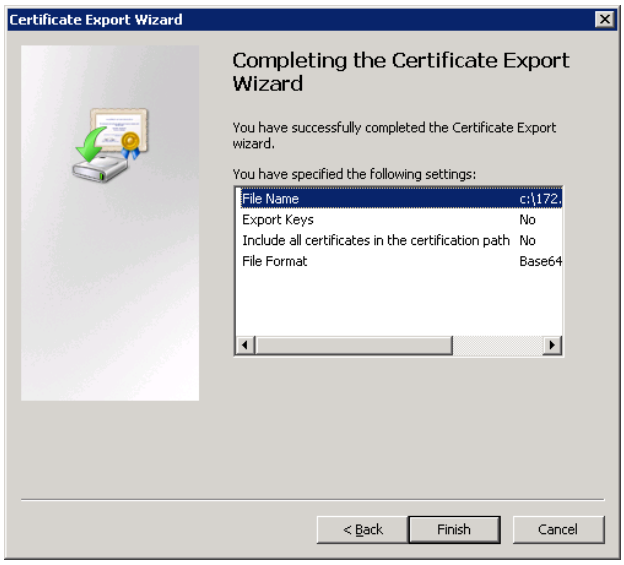


Make sure to select Base-64 encoded X.509. This is the PEM format required by NSE.

The CA certificate needs to be named <IP_Address_of_KMIP_Server>_CA.pem. For multiple KMIP servers, you would copy this file repeatedly and name them <IP_Address_of_KMIP_Server_1>_CA.pem, < IP_Address_of_KMIP_Server_2>_CA.pem, < IP_Address_of_KMIP_Server_3>_CA.pem, and so on.

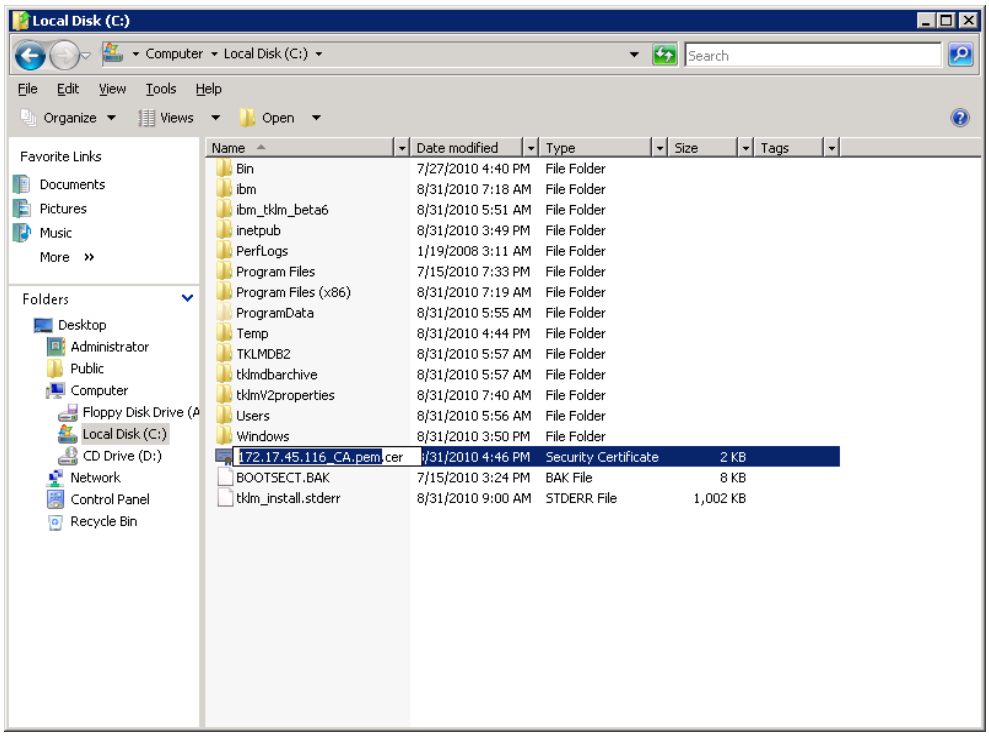
Figure 6-8: Completing Certificate Export Wizard

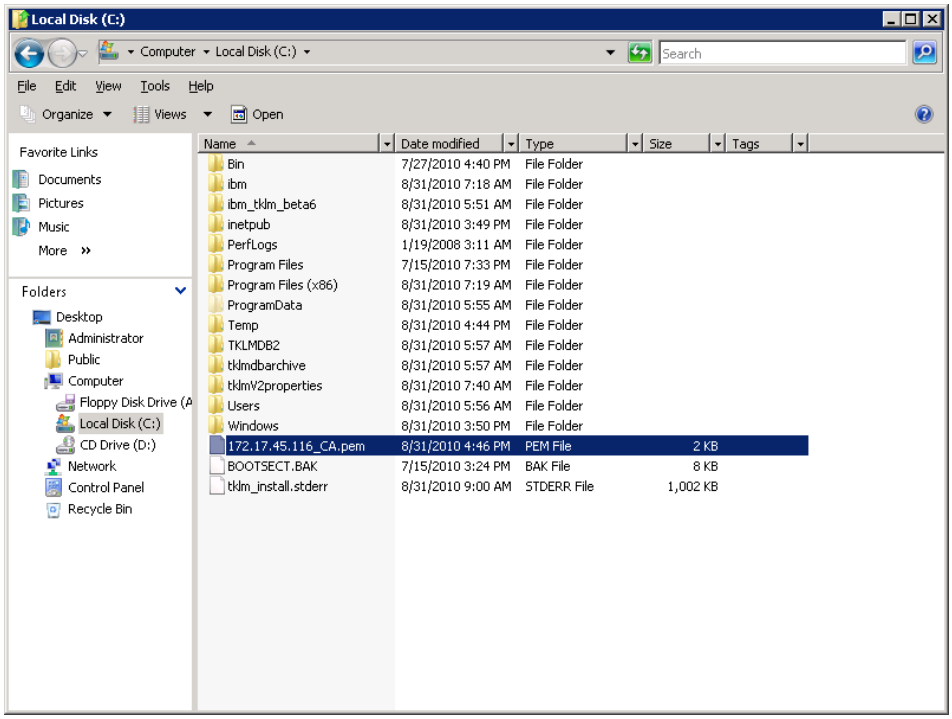




Rename the file to a .PEM extension. This extension is needed by NSE to properly recognize the certificate.

Figure 9-10: Renaming the .PEM file





You have now successfully created the public certificate from the certificate authority. This file will be

used in subsequent steps.

CONSIDERATIONS FOR HIERARCHICAL CA SERVERS

The preceding steps were for a single, standalone CA server. Many enterprises will have hierarchical CA servers: there will be a root CA at the top level and one or more subordinate CAs, sometimes forming a chain of trust. For environments where there is a chain of CA servers, the <IP_Address_of_KMIP_Server>_CA.pem file must concatenate the public certificates of each CA server in the chain.

For example, if a customer has three CA servers in a chain, Root_CA, Sub1_CA, Sub2_CA, you would need to obtain the public certificate from all three CA servers and concatenate them together:

```
cat Sub2_CA.pem Sub1_CA.pem > Sub1_Sub2_CA.pem
```

```
cat Sub1_Sub2_CA.pem Root_CA.pem > Root_CA_Sub1_Sub2_CA.pem
```

The resulting Root__Sub1_Sub2_CA.pem contains the public certificates of all three CA servers. This concatenated file would be renamed <IP_Address_of_KMIP_Server>_CA.pem.

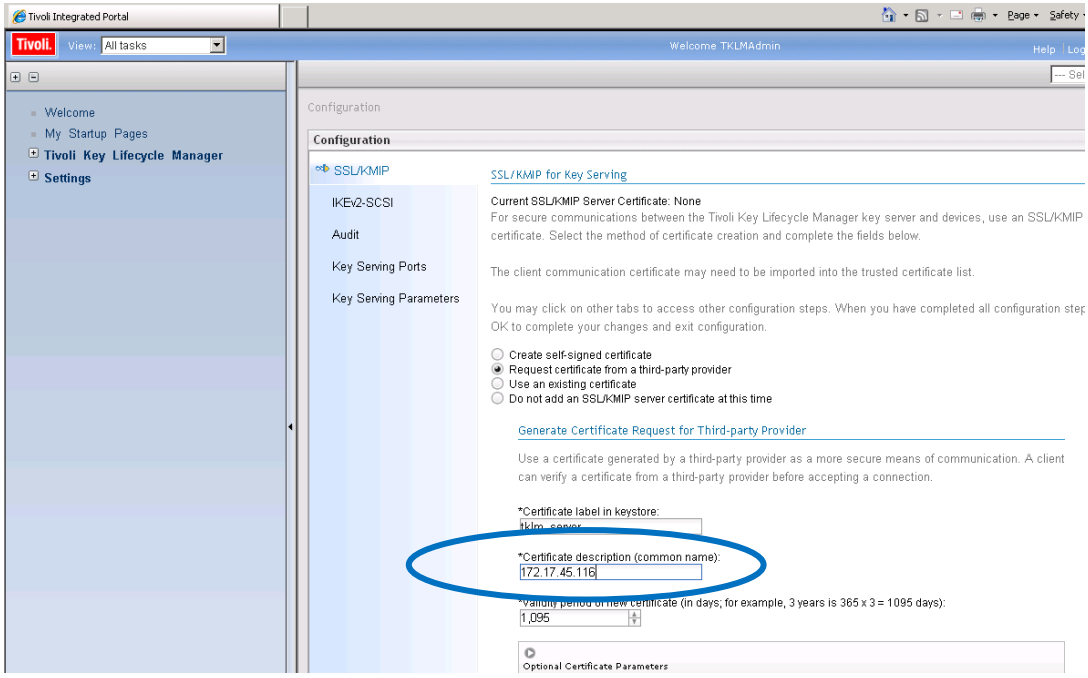
3.2 GENERATE AND EXPORT THE KMIP SERVER PUBLIC CERTIFICATE

Public and private certificates are needed for the KMIP server to establish trust with NSE. In this example, we will use IBM Tivoli Lifetime Key Management (TLKM) server v2 as our KMIP server.

Installation and configuration instructions for the IBM TKLMv2 can be found in the appendix. The following steps assume that the installation and configuration have already been completed.

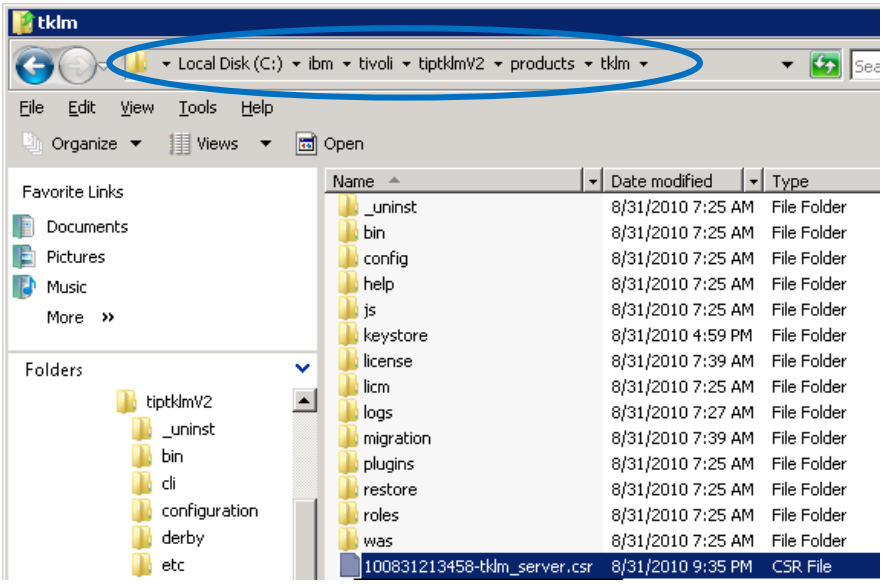
Generate the TKLM server public certificate and create a certificate signing request (.csr) file. Pay particular attention to the common name. You can enter either the IP address of the TKLM server or the DNS name. We will refer to it during the certificate export process.

Figure 11: Exporting the IBM TKLMv2 Public Certificate



The resulting file can be found at \<tklm install root>\tivoli\tiptklmV2\products\tklm\. This file will be sent to the CA for signing.

Figure 12: Location of Exported IBM TKLMv2 Public Certificate



3.3 GENERATE A PRIVATE AND PUBLIC KEY PAIR FOR NSE

This step needs to be done external to the NSE system. A public and private key pair can be generated in either Windows or UNIX using OpenSSL, but the following example shows how it's done using OpenSSL in Linux.

Generate the private key first.

```
root@core-vm30:~# openssl genrsa -des3 -out client_private.key 2048
Generating RSA private key, 2048 bit long modulus
..........+++
.....................................+++
e is 65537 (0x10001)
Enter pass phrase for client_private.key:
Verifying - Enter pass phrase for client_private.key:
```

The result will be a private key, as seen in the following example.

```
root@core-vm30:~# ls
client_private.key
```

Generate a certificate signing request (.csr) file from the private key. The file must be named client.csr.

```
root@core-vm30:~# openssl req -new -key client_private.key -out client.csr
Enter pass phrase for client_private.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) [AU]:Your Country
State or Province Name (full name) [Some-State]:Your State
Locality Name (eg, city) []:Your City
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Your Company
Organizational Unit Name (eg, section) []:Your OU
Common Name (eg, YOUR name) []:fas2040c-svl04.iops.eng.netap.com
Email Address []:your_email@your_company.com

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

The result will be a .csr file, which needs to be sent to the CA for signing. Note the preceding common name; we will refer to it later in our steps.

You now have two files: (1) a .csr file, which is the signing request for the public certificate for NSE, and (2) the client_private.key, which is the private key for NSE.

```
root@core-vm30:~# ls

client.csr  client_private.key
```

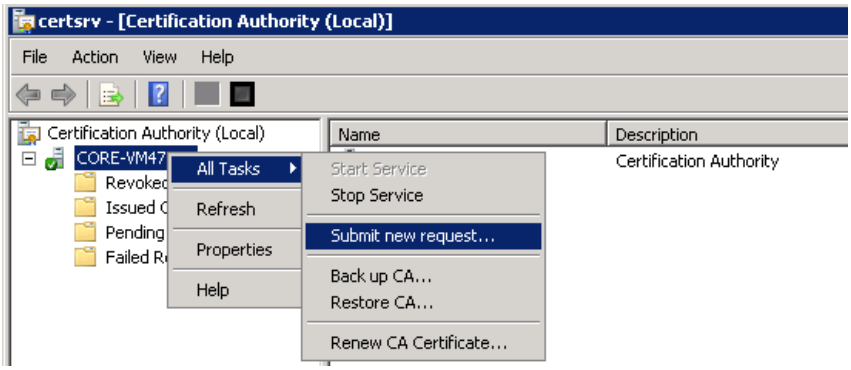
4 SIGN SSL CERTIFICATES AND EXPORT FOR USE

You should now have two .csr files: one from the TKLM server (in this example the file name is 100831213458-tklm_server.csr) and one generated using OpenSSL for NSE (in this example the file name is client.csr).

An important note for SSL certificates: Note the expiration time for all SSL certificates generated and make sure these are in line with your certificate expiration policies. When SSL certificates expire, new certificates will need to be generated and installed according to the procedures outlined in the following section. Failure to replace SSL certificates before expiration could result in an inability to retrieve data off the encrypted disks. For further information, refer to Appendix B: SSL Certificate Replacement.

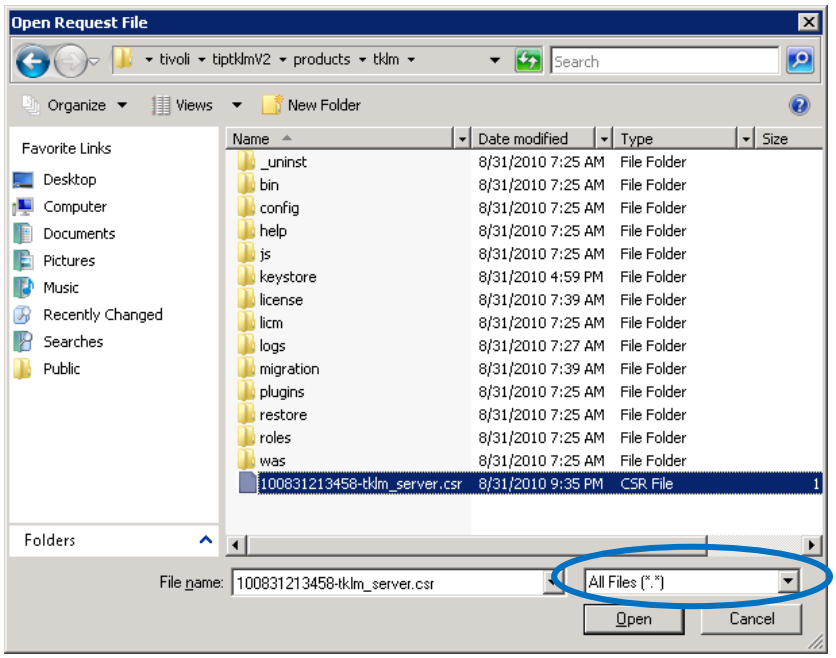
4.1 SIGNING THE TKLM .CSR FILE

Figure 13: Signing the CSR file in Windows 2008 CA



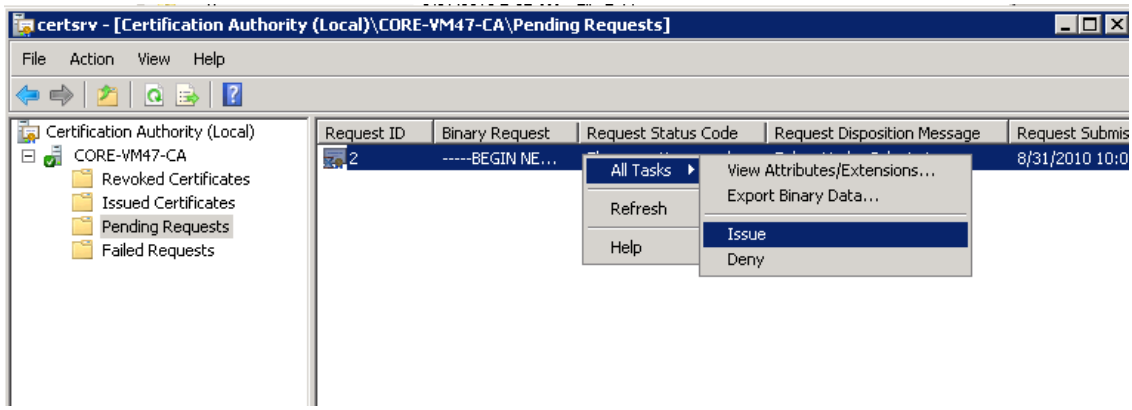
Select “All Files (*.*)” type to view the CSR file and browse to the correct folder, in this case \<tklm_install_root\tivoli\tiptklmV2\products\tklm\.

Figure 14: Locating the CSR file



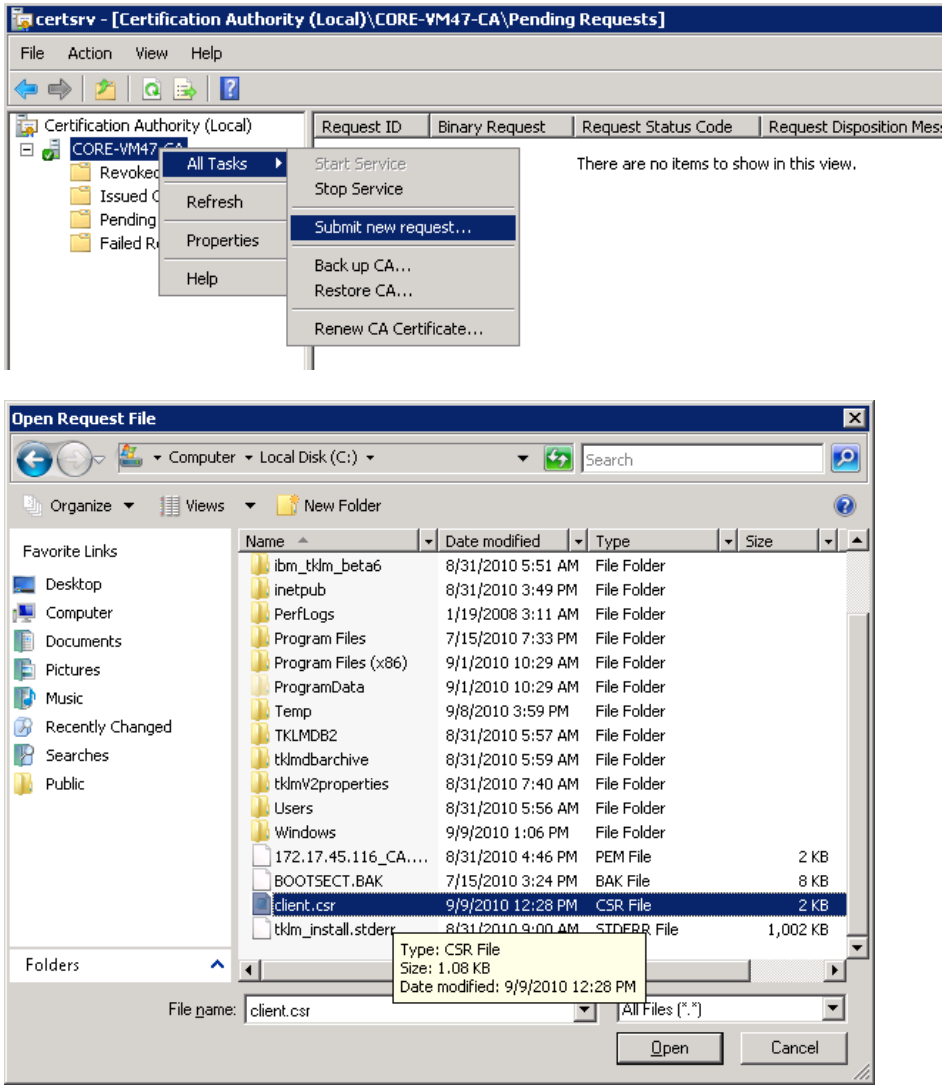
Select Pending Requests and sign the .csr file by selecting the “Issue” option.

Figure 15: Issuing the Certificate



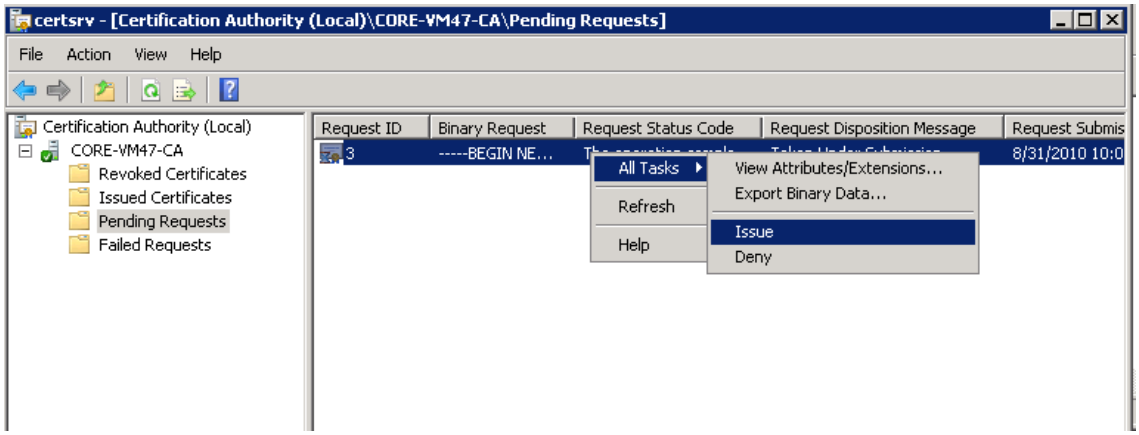
4.2 SIGNING THE CLIENT.CSR FILE

Figure 16-17: Submit CSR Request



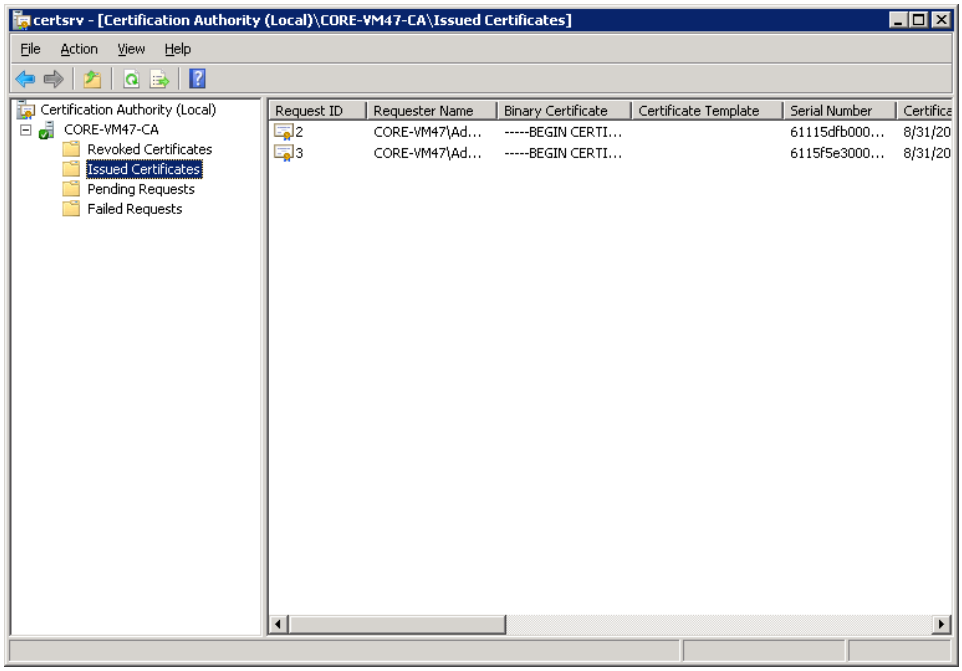
Select Pending Requests and sign the .csr file by selecting the “Issue” option.

Figure 18: Issue the Signed Certificate



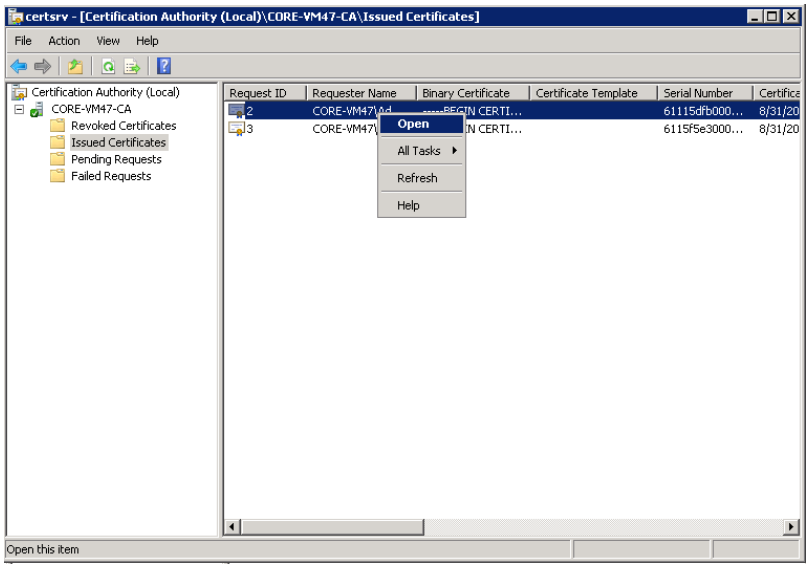
The result is two issued (signed) certificates in the CA.

Figure 19: Signed Certificate Result



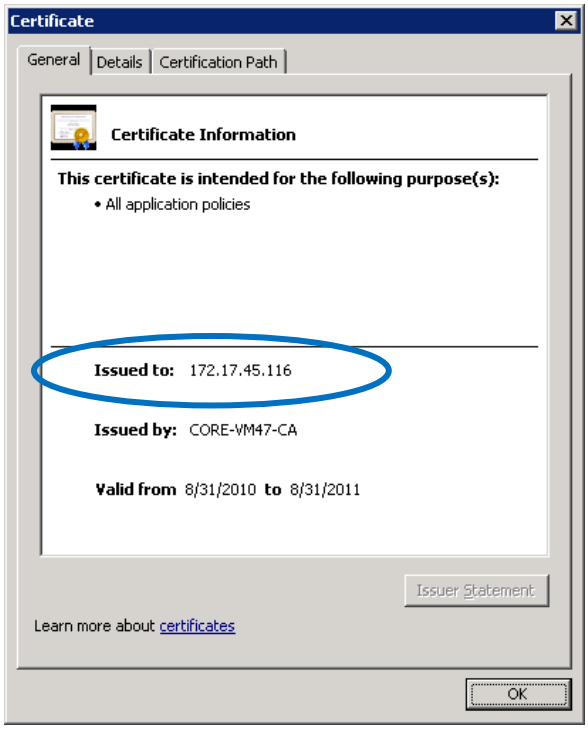
4.3 EXPORT THE SIGNED TKLM CERTIFICATE

Figure 20: Locate Certificate to be Exported

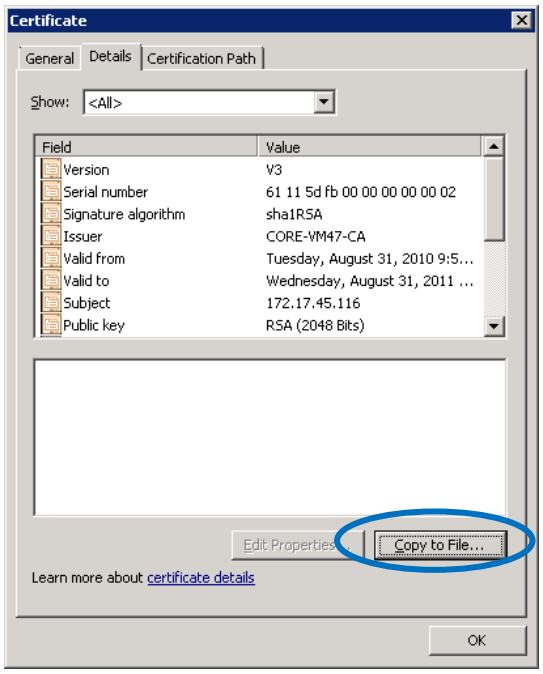


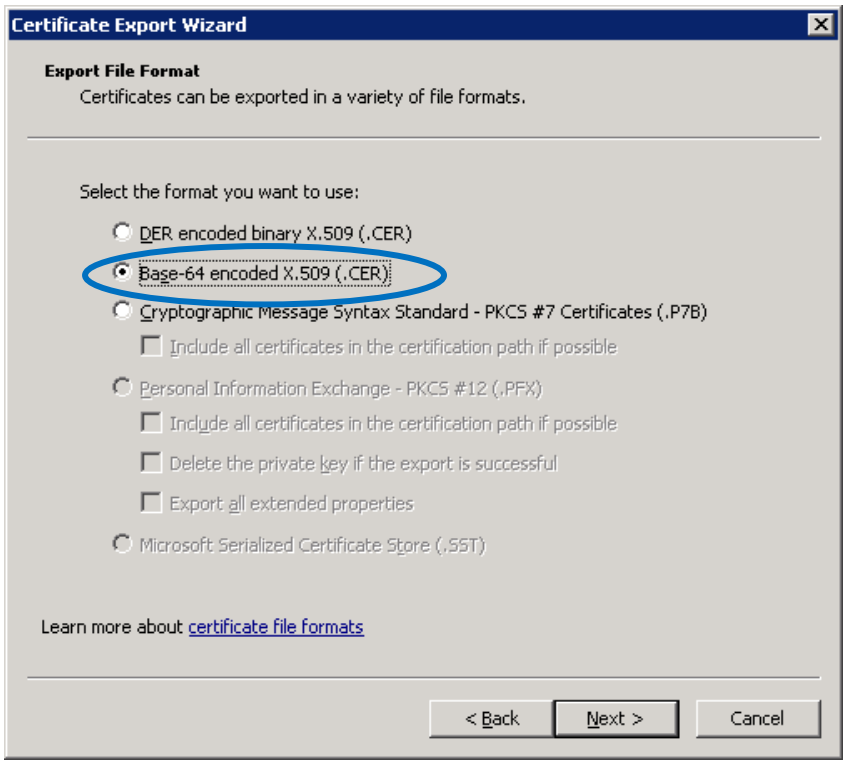
Confirm the TKLM server name under “Issued to:” This should match the common name during the CSR request from the TLKM server. In this case we used the IP address as the common name.

Figure 21: Confirm Correct IP Address for TKLM



Figures 22-24: Export Certificate Wizard

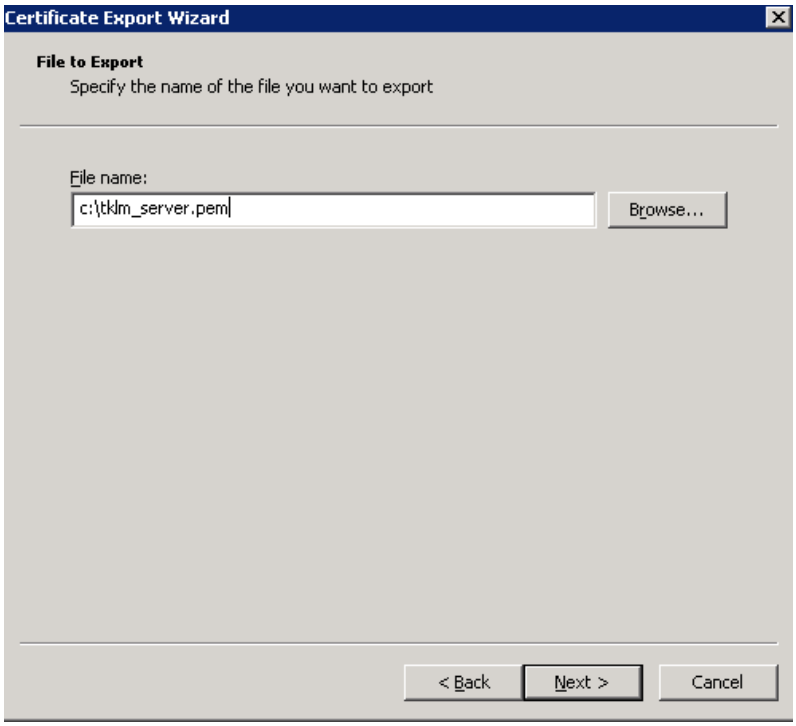


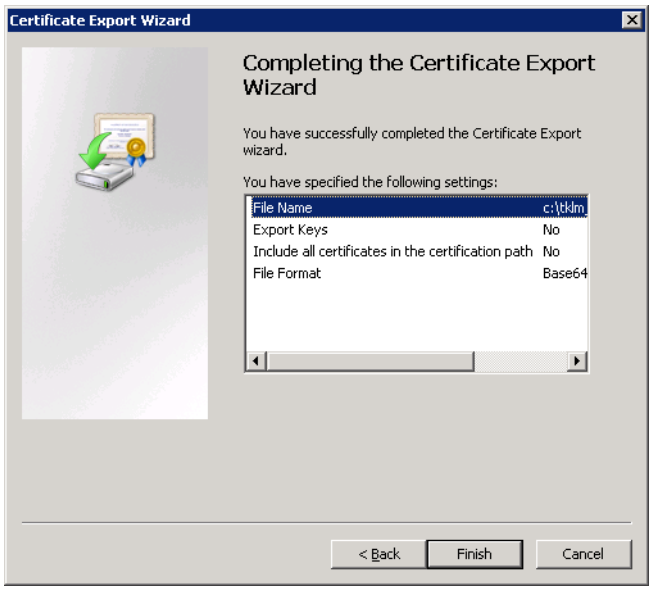


Make sure to select Base-64 encoded X.509. This is the PEM format required by NSE.

The file name here is not critical. This file will be imported back into the TKLM server.

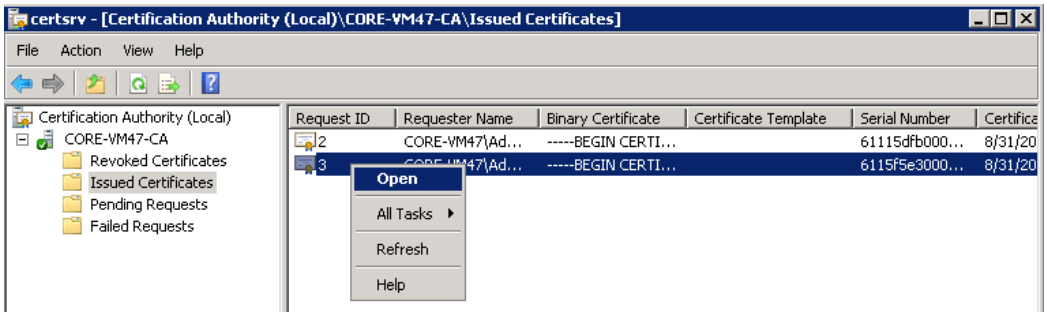
Figures 25-27: Name and Save Certificate File





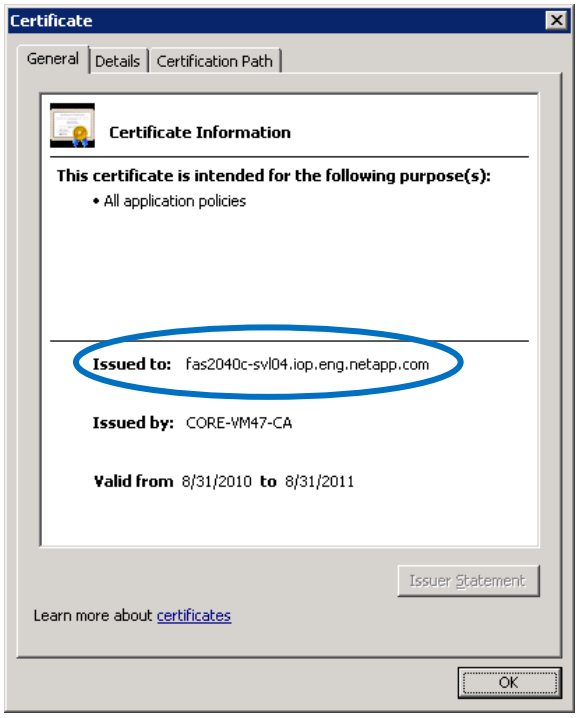
4.4 EXPORT THE SIGNED NSE CERTIFICATE

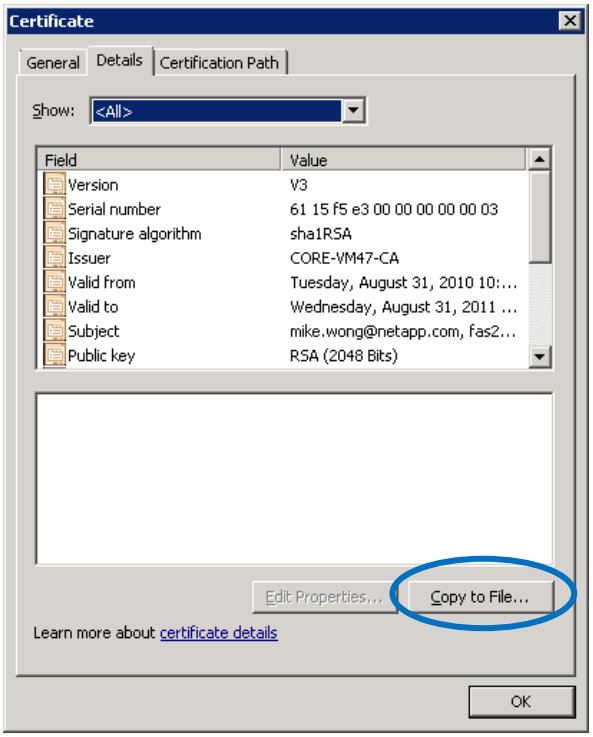
Figure 28: Locate and Export NSE Certificate



Make sure the chosen certificate matches the common name specified during the creation of the csr in OpenSSL.

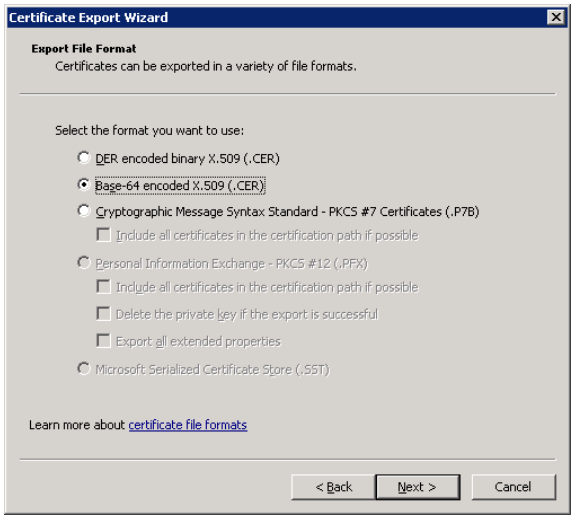
Figure 29-30: Confirm Common Name and Copy to File





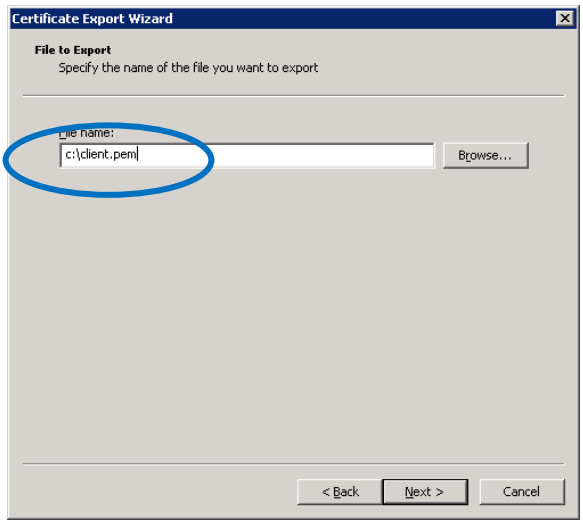
Figures 31-32: Certificate Export Wizard

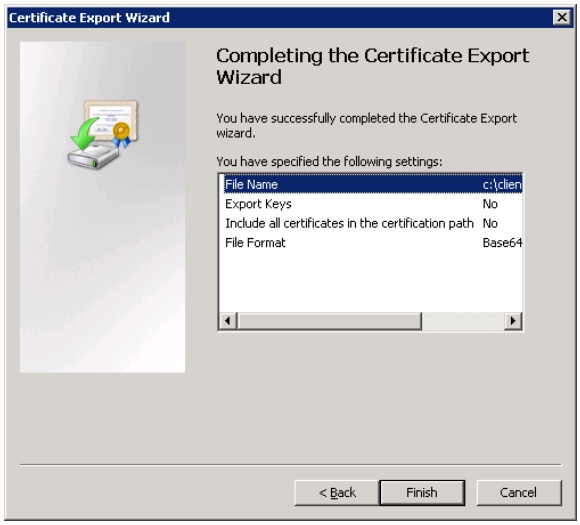
Make sure to select Base-64 encoded X.509. This is the PEM format required by NSE.



The file name here must be called “client.pem” in order to be used properly by NSE.

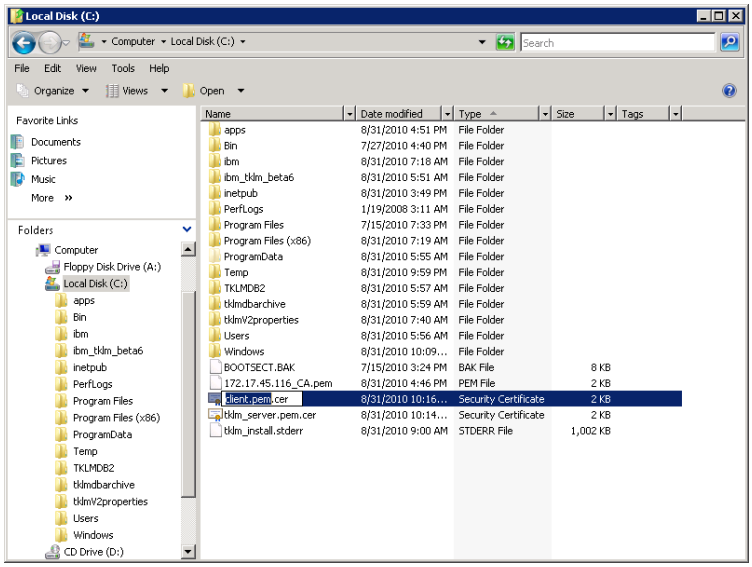
Figures 33-35: Rename CER file and Complete Export Wizard

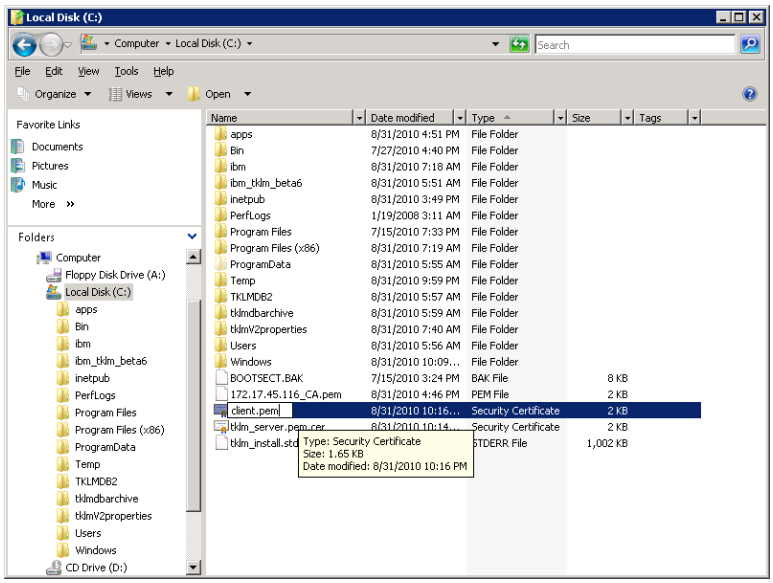


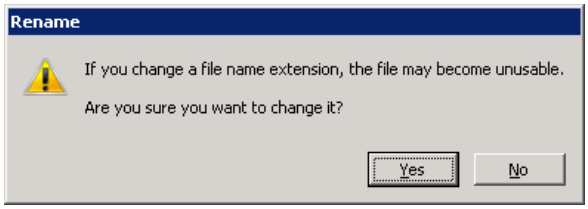


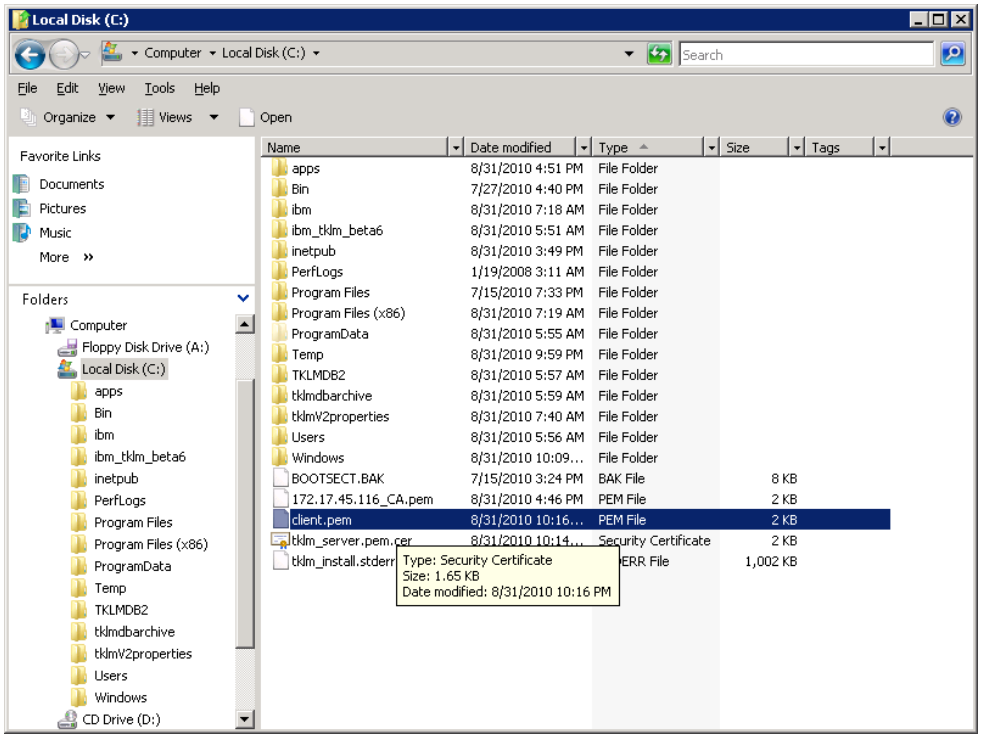
Remove the .cer extension from the client.pem file. NSE requires the file to have the .pem extension only.

Figures 36-39: Rename PEM file








Local Disk (C:)
Computer
Local Disk (C:)
Search
File
Edit
View
Tools
Help
Organize
Views
Open
Favorite Links
Documents
Pictures
Music
More
Folders
Computer
Floppy Disk Drive (A:)
Local Disk (C:)
apps
Bin
ibm
ibm_tklm_beta6
inetpub
PerfLogs
Program Files
Program Files (x86)
ProgramData
Temp
TKLMDB2
tklmdbarchive
tklmV2properties
Users
Windows
CD Drive (D:)
Name
Date modified
Type
Size
Tags
apps 8/31/2010 4:51 PM File Folder
Bin 7/27/2010 4:40 PM File Folder
ibm 8/31/2010 7:18 AM File Folder
ibm_tklm_beta6 8/31/2010 5:51 AM File Folder
inetpub 8/31/2010 3:49 PM File Folder
PerfLogs 1/19/2008 3:11 AM File Folder
Program Files 7/15/2010 7:33 PM File Folder
Program Files (x86) 8/31/2010 7:19 AM File Folder
ProgramData 8/31/2010 5:55 AM File Folder
Temp 8/31/2010 9:59 PM File Folder
TKLMDB2 8/31/2010 5:57 AM File Folder
tklmdbarchive 8/31/2010 5:59 AM File Folder
tklmV2properties 8/31/2010 7:40 AM File Folder
Users 8/31/2010 5:56 AM File Folder
Windows 8/31/2010 10:09... File Folder
BOOTSECT.BAK 7/15/2010 3:24 PM BAK File 8 KB
172.17.45.116_CA.pem 8/31/2010 4:46 PM PEM File 2 KB
client.pem 8/31/2010 10:16... PEM File 2 KB
tklm_server.pem.cer 8/31/2010 10:14... Security Certificate 2 KB
tklm_install.stderr ERR File 1,002 KB
Type: Security Certificate
Size: 1.65 KB
Date modified: 8/31/2010 10:16 PM

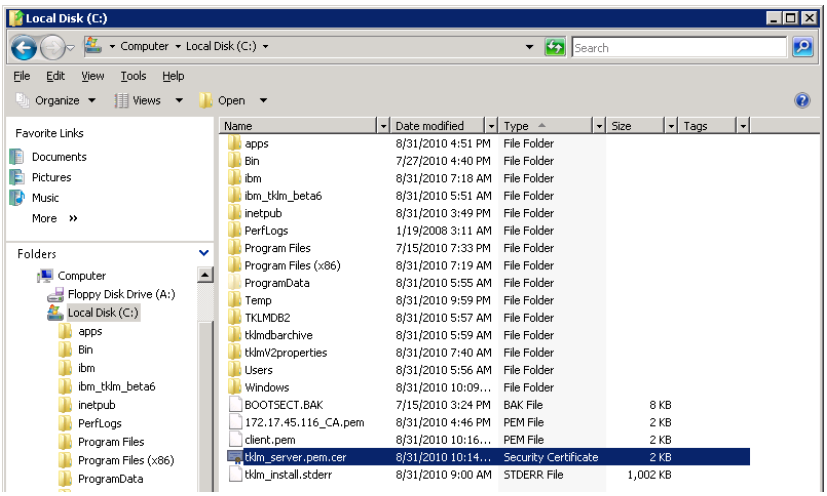
5 IMPORT THE SIGNED SSL CERTIFICATES

You now have three SSL certificates in all: the public certificate from the CA server and the two signed certificates. This section covers import of the SSL certificates into their correct locations on the KMIP server (TLKMv2 in this example) and on NSE.

5.1 IMPORT THE SIGNED TKLM CERTIFICATE BACK INTO THE TKLM SERVER

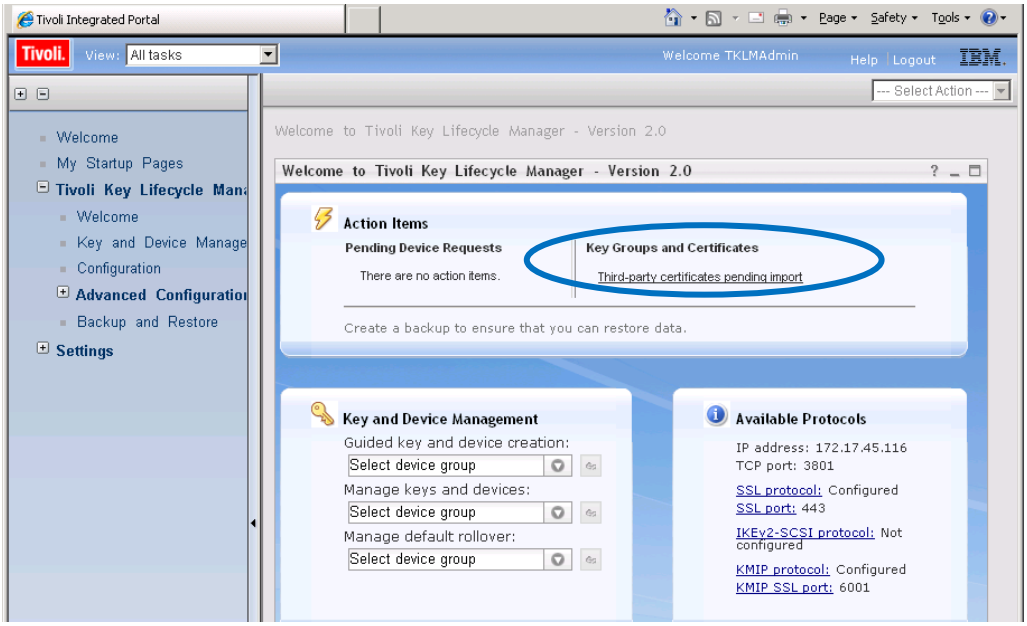
The TKLM server does not require the signed server certificate to have a specific file name format or extension. It does not need to be renamed prior to import.

Figure 40: Locate TKLM Certificate for Import



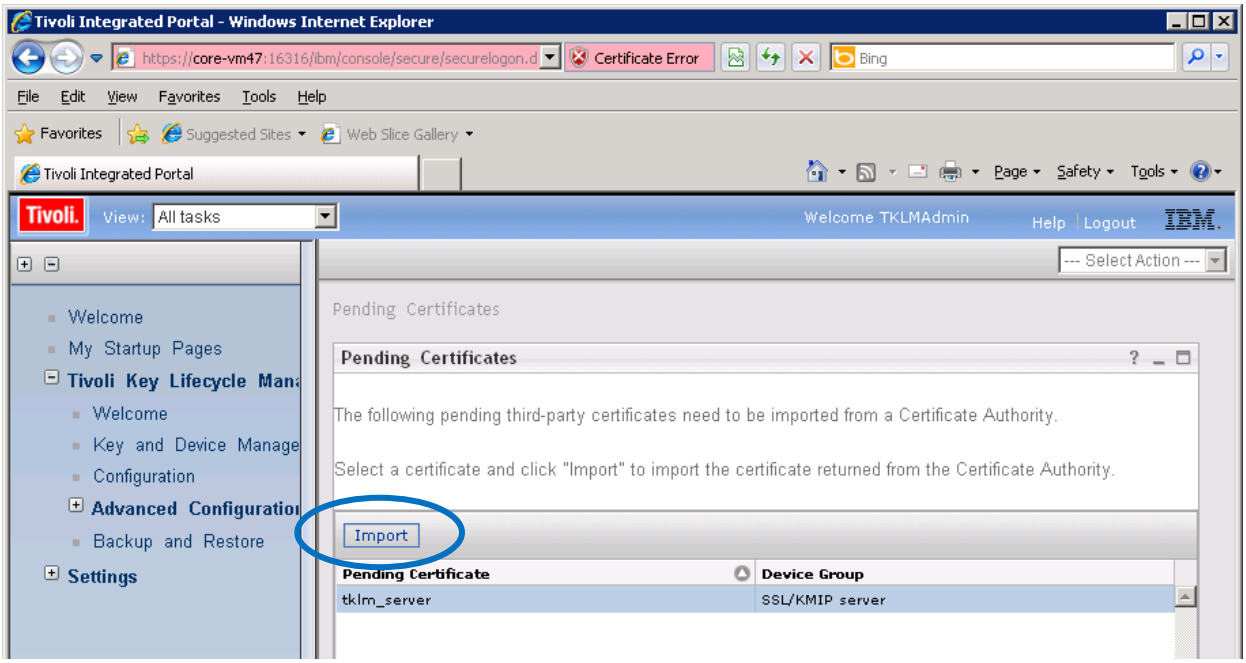
Log back into the TKLM portal and select the Welcome page.

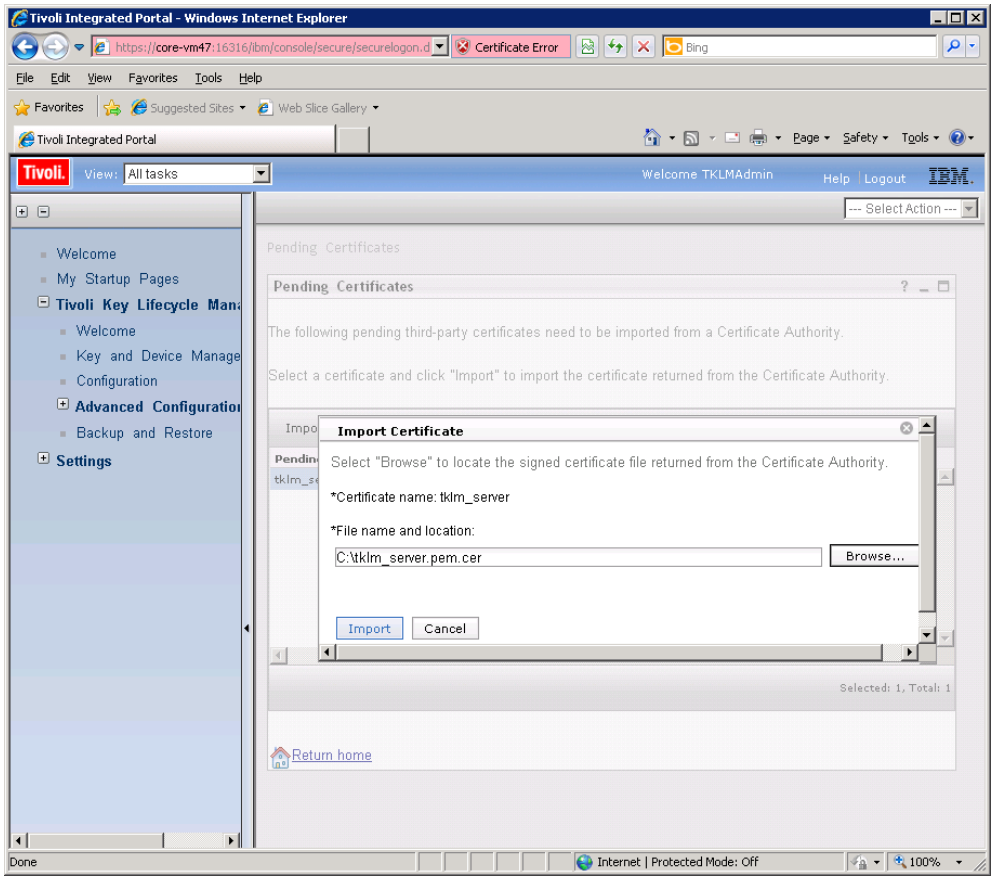
Figure 41: Third Party Certificates to be Imported

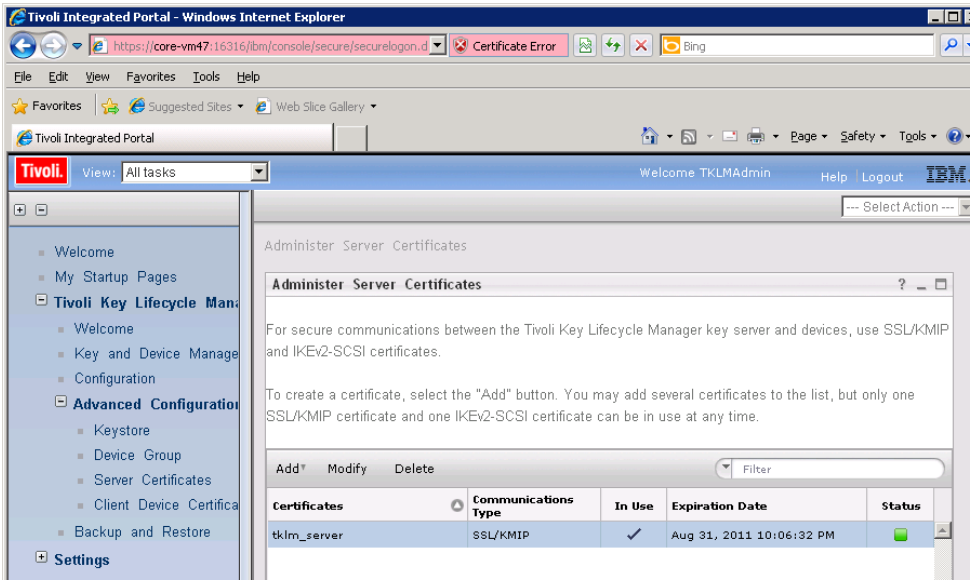


Select Import and browse to the location of the signed TKLM server certificate.

Figures 42-44: Import Signed TKLM Certificate





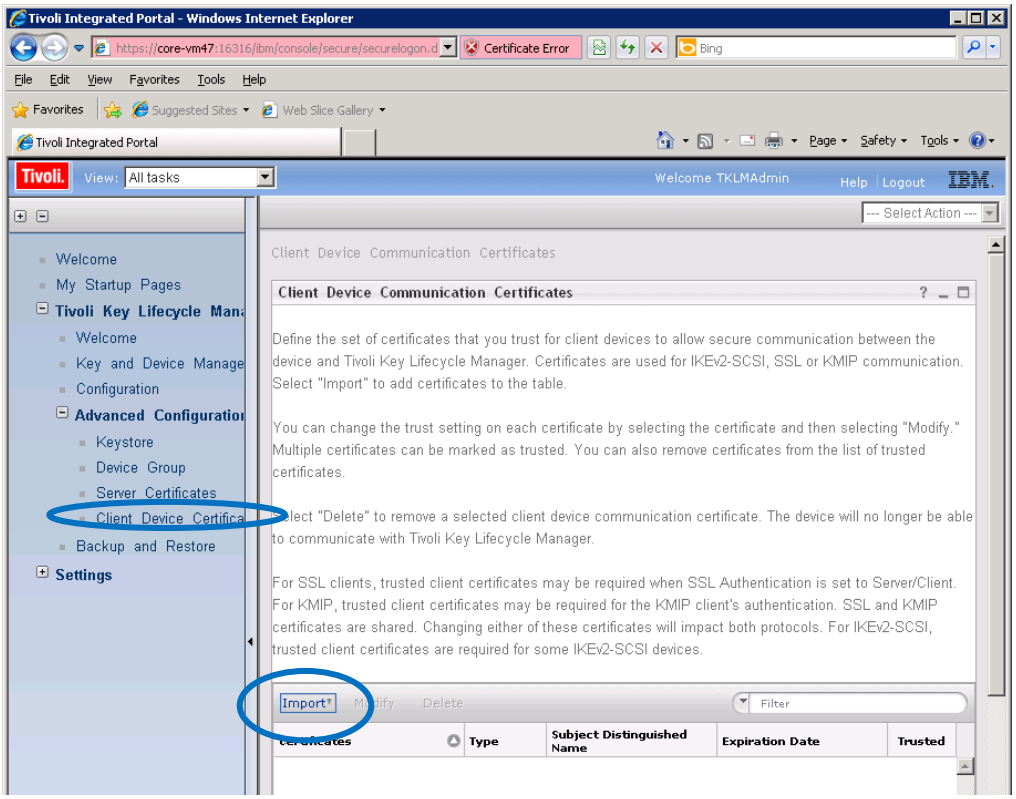


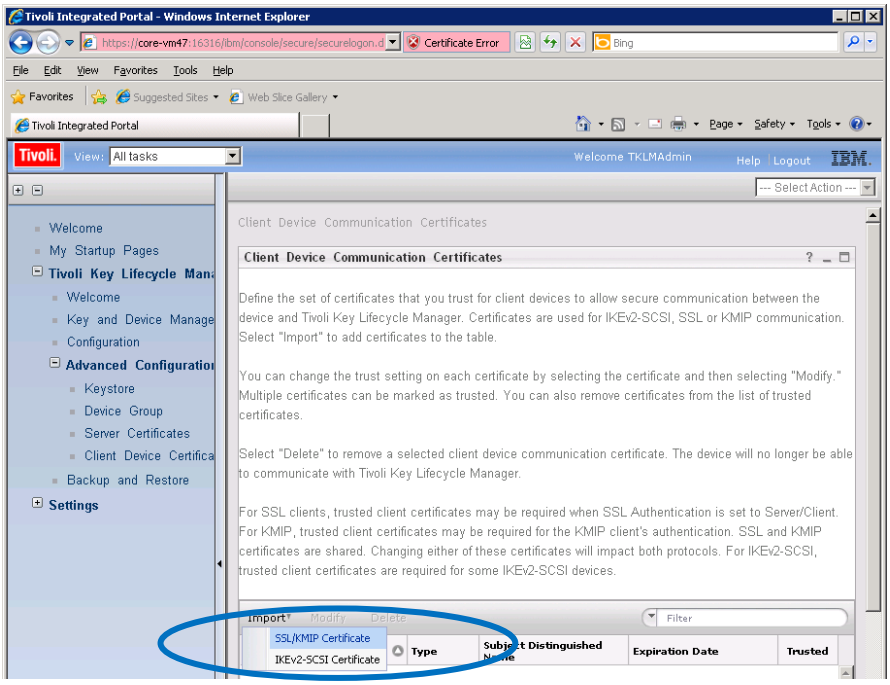
You have now successfully imported the signed TKLM public certificate into the TKLM server.

5.2 IMPORT CLIENT DEVICE CERTIFICATE

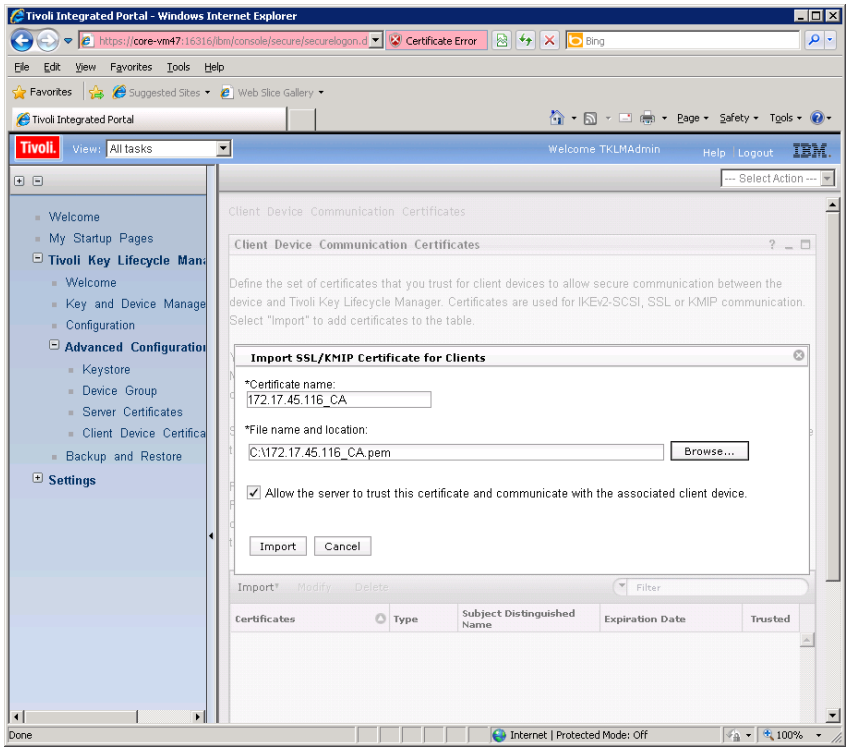
Next we need to import the CA certificate used to sign the NSE public certificate into the TKLM server. This might or might not be from the same CA used to sign the TKLM public certificate. In this example, we are using the same CA to sign both certificates.

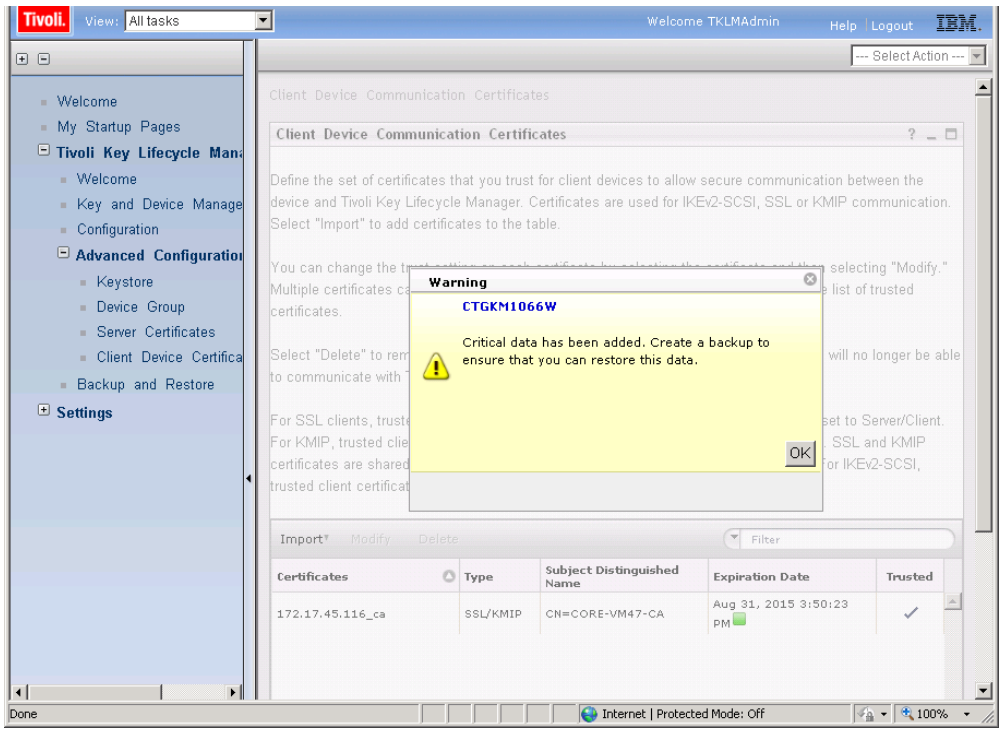
Figure 45-49: Import Client Certificate

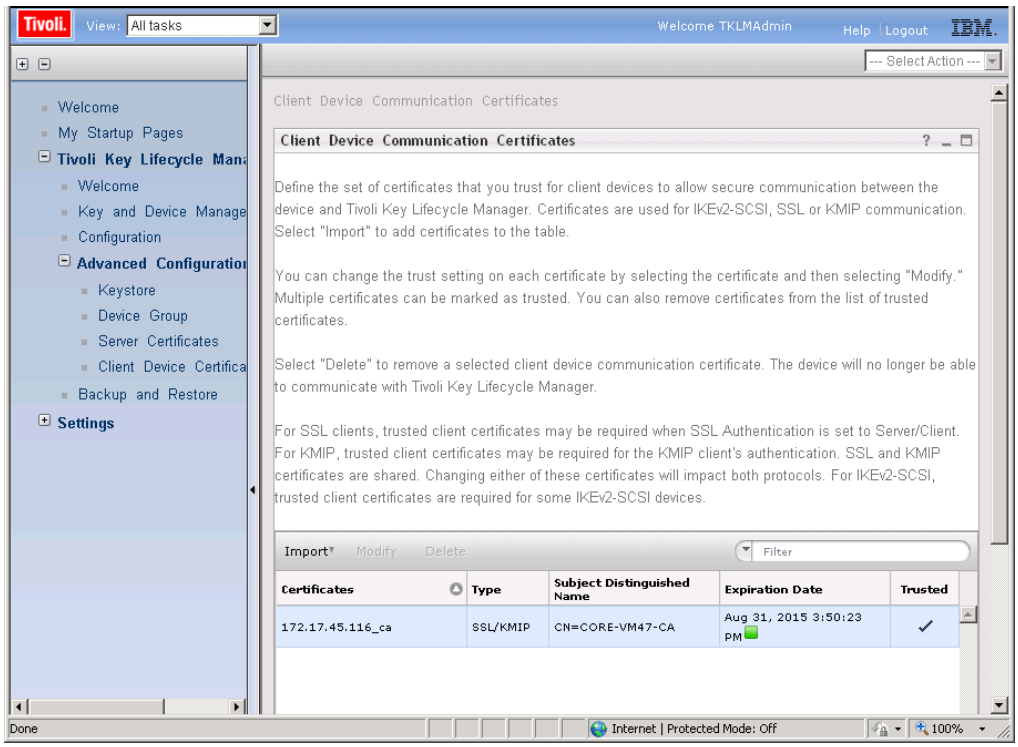




Name the certificate <IP_Address_of_KMIP_Server>_CA and browse to the <IP_Address_of_KMIP_Server>_CA.pem file.







You have successfully installed the CA certificate used to sign the NSE public certificate onto the TKLM server.

5.3 IMPORT SSL CERTIFICATES INTO NSE

CREATE PEM FILES FOR NSE

Next you need to create the client_private.pem file, which is the PEM format of the NSE private key. This file is needed by NSE to complete the setup.

Remove the passphrase used to protect the private key (this step is optional, but is shown for convenience).

```
root@core-vm30:~# mv client_private.key client_private.key.orig
root@core-vm30:~# openssl rsa -in client_private.key.orig -out
client_private.key
Enter pass phrase for client_private.key.orig:
writing RSA key
```

Create the client_private.pem file by concatenating the contents of the client.pem file into the client_private.key file.

```
root@core-vm30:~# cat client.pem client_private.key > client_private.pem
root@core-vm30:~# ls
client.csr  client.pem  client_private.key  client_private.key.orig
client_private.pem
root@core-vm30:~#
```

You have now successfully created all the necessary keys and certificate files needed by NSE.

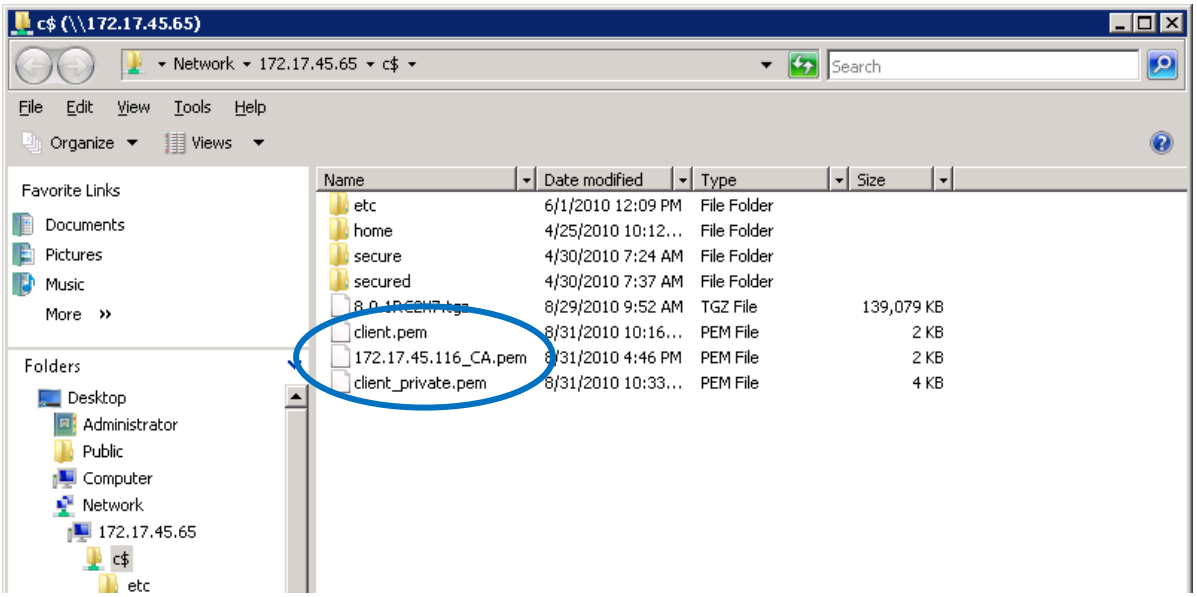
COPY KEYS AND CERTIFICATES TO NSE

Once all files have been properly signed and created, the following three files need to be copied onto the FAS platform, which has NSE drives installed. Make note of the path for these files. In this example \\172.17.45.65\c$ corresponds to 172.17.45.65:/vol/vol0/.

The three files are:

- client.pem: This file is the NSE client signed public key in PEM format. This file was generated earlier using OpenSSL and signed by the CA.
- client_private.pem: This file is the NSE client private key in PEM format. This file was generated as one of the last steps after the client.pem file was generated.
- 172.17.45.116_CA.pem: This is the exported CA certificate used to sign the TKLM server public certificate. This might or might not be the same file used to sign the NSE public certificate. In this example, we used a single CA to sign both files.

Figure 50: Moving PEM Files on NSE Volume



6 VERIFICATION OF PEM FILES

Verification of PEM files for a given configuration can be performed from any computer that has OpenSSL installed and is able to communicate with the KMIP server. Copy the three files—client.pem, <IP_Address_of_KMIP_Server>_CA.pem, and the client_private.pem—to a computer running OpenSSL and issue the following command:

```
openssl s_client -tls1 -connect < IP_Address_of_KMIP_Server >:6001 -verify 10
-showcerts -cert client.pem -key client_private.pem -CAfile
<IP_Address_of_KMIP_Server>_CA.pem
```

The following is an example of normal output:

```
[root@ts-pepato NSE]# openssl s_client -tls1 -connect 10.42.152.38:6001
-verify 10 -showcerts -cert client.pem -key client_private.pem -CAfile
10.42.152.151_CA.pem

verify depth is 10

CONNECTED(00000003)

depth=1 /DC=com/DC=netapp/DC=vm-w2k8r2-user/CN=vm-w2k8r2-user-TS-ENTERPRISE-
CA

verify return:1

depth=0 /CN=tklm-server

verify return:1

---

Certificate chain
```

```
 0 s:/CN=tklm-server
   i:/DC=com/DC=netapp/DC=vm-w2k8r2-user/CN=vm-w2k8r2-user-TS-ENTERPRISE-CA
-----BEGIN CERTIFICATE-----
MIIFTTCCBDWgAwIBAgIKXKi+UAAAAAAACDANBgkqhkiG9w0BAQUFADB7MRMwEQYK
~~~~~~~~~~~~~~SNIP CERTIFICATE CONTENTS~~~~~~~~~~~~~~~~~~~~~~~~~~
WPXcwwDyViK9AuRMK84QERE=
-----END CERTIFICATE-----
---
Server certificate
subject=/CN=tklm-server
issuer=/DC=com/DC=netapp/DC=vm-w2k8r2-user/CN=vm-w2k8r2-user-TS-ENTERPRISE-CA
---
Acceptable client certificate CA names
/DC=com/DC=netapp/DC=vm-w2k8r2-user/CN=vm-w2k8r2-user-TS-ENTERPRISE-CA
/C=US/ST=CA/L=Sunnyvale/O=NetApp/OU=NGS/CN=ts-
scrooge/emailAddress=user@netapp.com
/C=US/ST=CA/L=Costa Mesa/O=Emulex Corp/OU=Emulex Mfg/CN=OneSecure
/C=US/O=ibmTapeDrive
/C=US/O=ibmTapeDrive
/C=US/O=IBM/CN=subca
/C=US/CN=DS5000StorageRoot/OU=StorageHardware/O=ibmDiskDrive
---
SSL handshake has read 2656 bytes and written 3028 bytes
---
New, TLSv1/SSLv3, Cipher is EDH-RSA-DES-CBC3-SHA
Server public key is 2048 bit
SSL-Session:
    Protocol  : TLSv1
    Cipher    : EDH-RSA-DES-CBC3-SHA
    Session-ID:
4CDB30DCE046F75D16076BD238B73ED0DEC6BBBE8D4D747197FF6F005F747B7A
    Session-ID-ctx:
    Master-Key:
F5A3A376B8938F010FA27C5400B384B49F012D9AF77F08959BD47A8AF7324EF9F3FA5D10E0A70
9FB320F85BB28170929
    Key-Arg   : None
    Krb5 Principal: None
    Start Time: 1289433073
    Timeout   : 7200 (sec)
```

```
    Verify return code: 0 (ok)
---
```

At this point, the system will wait for a break (CONTROL-C) to return to a shell prompt. The highlighted section is the important output. This is an example of successful output using three valid PEM files.

7 HA CLUSTER PAIR SSL CERTIFICATE CONSIDERATIONS

On any HA system with two controllers, certificates must be copied to the other controller and installed. To make sure of proper disk failover for HA systems, the same client-side certificates and private keys must be used on each controller. Simply take the three files created earlier for the first controller and copy them to the equivalent location on the second controller.

Once you have completed this guide, you are ready to run through the NSE setup. Refer to the document "Data ONTAP 8.1 7-Mode Software Setup Guide" and turn to the section on storage encryption to complete the initialization and configuration of NSE.

APPENDIXES

APPENDIX A: CERTIFICATE CLEANUP

A.1 DELETION OF CERTIFICATES

Prior to installation of new certificates, old certificates must be removed to make sure the updated certificates are used. The following procedure should be followed to completely remove all certificates on an NSE system.

1. Remove the key management server:
   ```
   a. key_manager remove –key_server <IP_Address_of_KMIP_Server>
   ```
2. Remove all certs installed by NSE:
   ```
   a. keymgr delete cert client_private.pem
   b. keymgr delete cert client.pem
   c. keymgr delete cert <IP_Address_of_KMIP_Server>_CA.pem
   ```

Once this has been executed, you are ready to reinstall with new certificates.

Note: If you are running clustered Data ONTAP 8.2, the above NSE commands should be executed at the nodeshell level. They are not available in clustershell. You can access the nodeshell in clustered Data ONTAP by running "system node run –node nodename" at the clustershell level. Once you are into the nodeshell, you can execute the corresponding 7-Mode NSE command.

APPENDIX B: SSL CERTIFICATE REPLACEMENT

SSL certificates all have expiration periods after initial creation. After a predetermined time, the certificates will no longer be valid and should be replaced in advance of the expiration date. Certificates used in NSE can be queried for their expiration dates using the following command:

```
keymgr view cert client.pem
```

Replacement of SSL certificates should use the following procedure:

1. Generate new SSL certificates and sign them:
   a. Follow preceding procedures in sections 1-3 to generate and sign your new certificates.
2. Rekey all encrypted drives to 0x0 temporarily:
   a. disk encrypt rekey 0x0 *
   This step can take a few seconds to a few minutes to complete, depending on the number of drives installed in the system. During this time, drives will not have an authentication key associated to them, so care must be exercised to make sure of physical security of the drives.
3. Uninstall all existing certificates according to steps in Appendix A: Certificate Cleanup.
4. Install new certificates according to section 5: Import the Signed SSL Certificates.
5. Verify connectivity using the new certificates:
   a. key_manager query
   You should receive successful output of existing key IDs.
6. Rekey all drives using “key_manager rekey”:
   a. You can use –manual to specify your own authentication key or allow Data ONTAP to generate one automatically for you.
   b. This process will take several seconds to several minutes depending on the number of drives you have installed.

Note: If you are running clustered Data ONTAP 8.2, the above NSE commands should be executed at the nodeshell level. They are not available in clustershell. You can access the nodeshell in clustered Data ONTAP by running “system node run –node nodename” at the clustershell level. Once you are into the nodeshell, you can execute the corresponding 7-Mode NSE command.

APPENDIX C: SELF-SIGNED CERTIFICATES

Self-signed certificates can be used in place of CA signed certificates. This might be advantageous for those who don’t have access to a certificate authority, but management of the individual certificates for multiple systems can be more cumbersome.

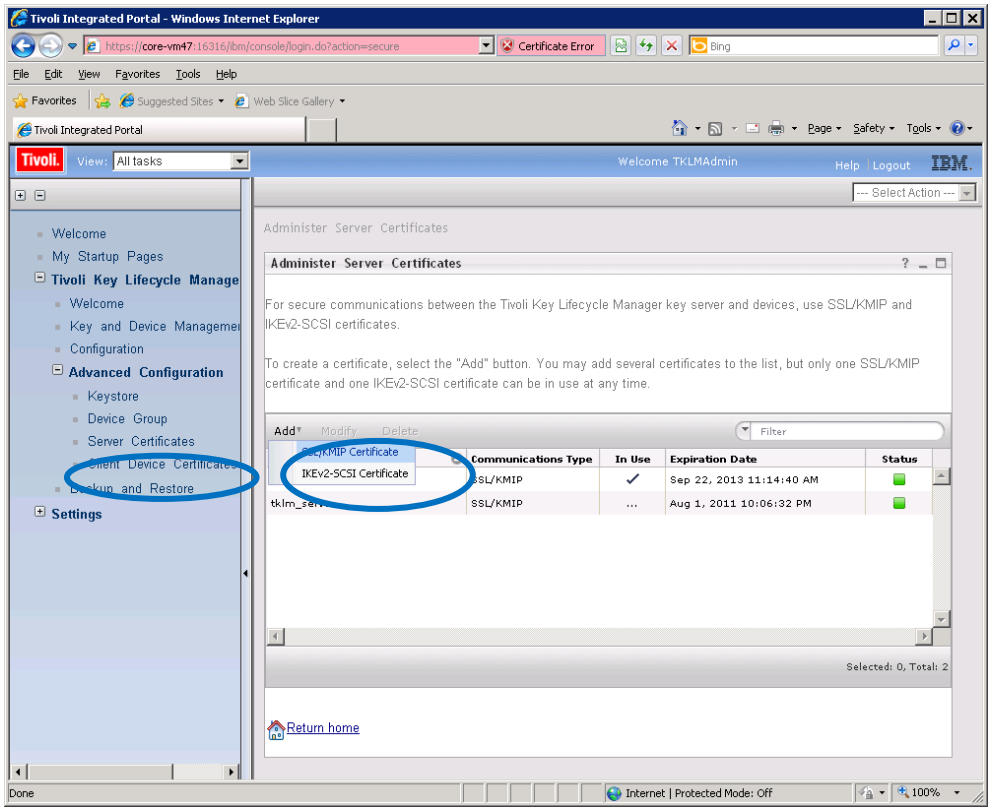
If a common CA is used to sign all certificates, this single CA certificate can be imported at each NSE client and each KMIP server. This reduces the number of certificates to keep track of and import. For self-signed certificates, the NSE client will need to import separate certificates from each KMIP server (up to four at this time), and each KMIP server will need to import separate certificates from each NSE client (which could be many depending on the size of the deployment).

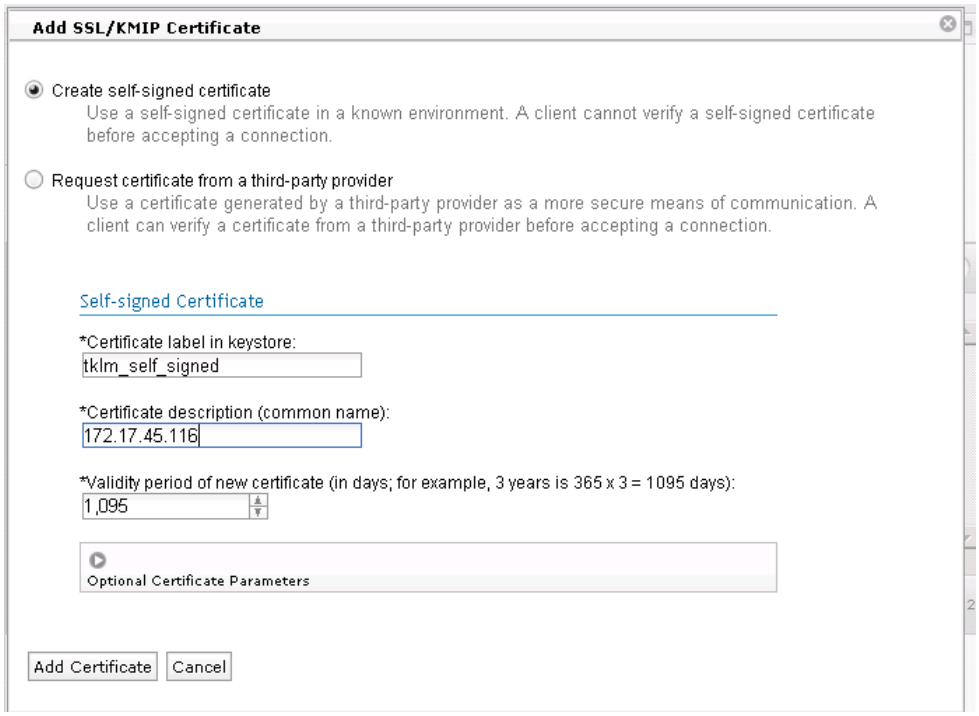
This appendix will attempt to illustrate how using self-signed certificates will affect the preceding detailed steps.

C.1 GENERATION OF A TKLM SELF-SIGNED CERTIFICATE

On the TKLM server, select Server Certificates and Add an SSL/KMIP Certificate.

Figures 51-52: Generate TKLM Self Signed Certificate





The certificate description here (common name) is not critical. In this example, we are using the IP address of the KMIP server, but it can also be the hostname of the KMIP server.

C.2 EXPORTING THE TLKM SELF-SIGNED CERTIFICATE

Unfortunately, this version of IBM TKLMv2 does not support export of certificates using the GUI; therefore, this step must be performed using a CLI.

Open a COMMAND window and execute the following three steps.

LOG IN TO THE TKLM CLI INTERFACE

```
C:\ibm\tivoli\tiptklmV2\bin>wsadmin -username tklmadmin -password
yourpassword -lang jython

WASX7209I: Connected to process "server1" on node TIPNode using SOAP
connector;

 The type of process is: UnManagedProcess

WASX7031I: For help, enter: "print Help.help()"
```

DISPLAY ALL CERTIFICATES KNOWN TO TKLM

```
wsadmin>print AdminTask.tklmCertList()

CTGKM0001I Command succeeded.


uuid = CERTIFICATE-a3f9ba28-ef1d-4509-b973-66f0a2ba6222

alias = tklm_server

key store name = defaultKeyStore

key state = ACTIVE

issuer name = CN=CORE-VM47-CA

subject name = CN=172.17.45.116

creation date = 8/31/10 9:34:59 PM PDT

expiration date = 8/31/11 10:06:32 PM PDT

serial number = 458389915494073329778690


uuid = CERTIFICATE-d8ec8612-5091-4611-8418-4d504e670870

alias = tklm_self_signed

key store name = defaultKeyStore

key state = ACTIVE

issuer name = CN=172.17.45.116

subject name = CN=172.17.45.116

creation date = 9/23/10 11:14:41 AM PDT

expiration date = 9/22/13 11:14:40 AM PDT
```

```
serial number = 831667365572719
```

EXPORT THE CERTIFICATE WITH THE CORRECT UUID

```
wsadmin>print AdminTask.tklmCertExport ('[-uuid CERTIFICATE-d8ec8612-5091-
4611-8418-4d504e670870 -format base64 -fileName c:\\tklm_self_signed.pem]')
CTGKM0001I Command succeeded.
c:\tklm_self_signed.pem
```

Note that the export command needs to specify the UUID and the base64 format. The output file created is “tklm_self_signed.pem." Remember that you will need to rename this file to <IP_Address_KMIP_SERVER.pem> prior to copying it to the NSE system.

C.3 CREATE THE NSE SELF-SIGNED CERTIFICATE

Section 5 should be followed as stated to generate the NSE CSR file and NSE private key.

Section 10 should also be followed to remove the passphrase from the private key.

CREATING THE NSE SELF-SIGNED CERTIFICATE

Instead of sending the NSE CSR file to the CA, the CSR can be self-signed using the following:

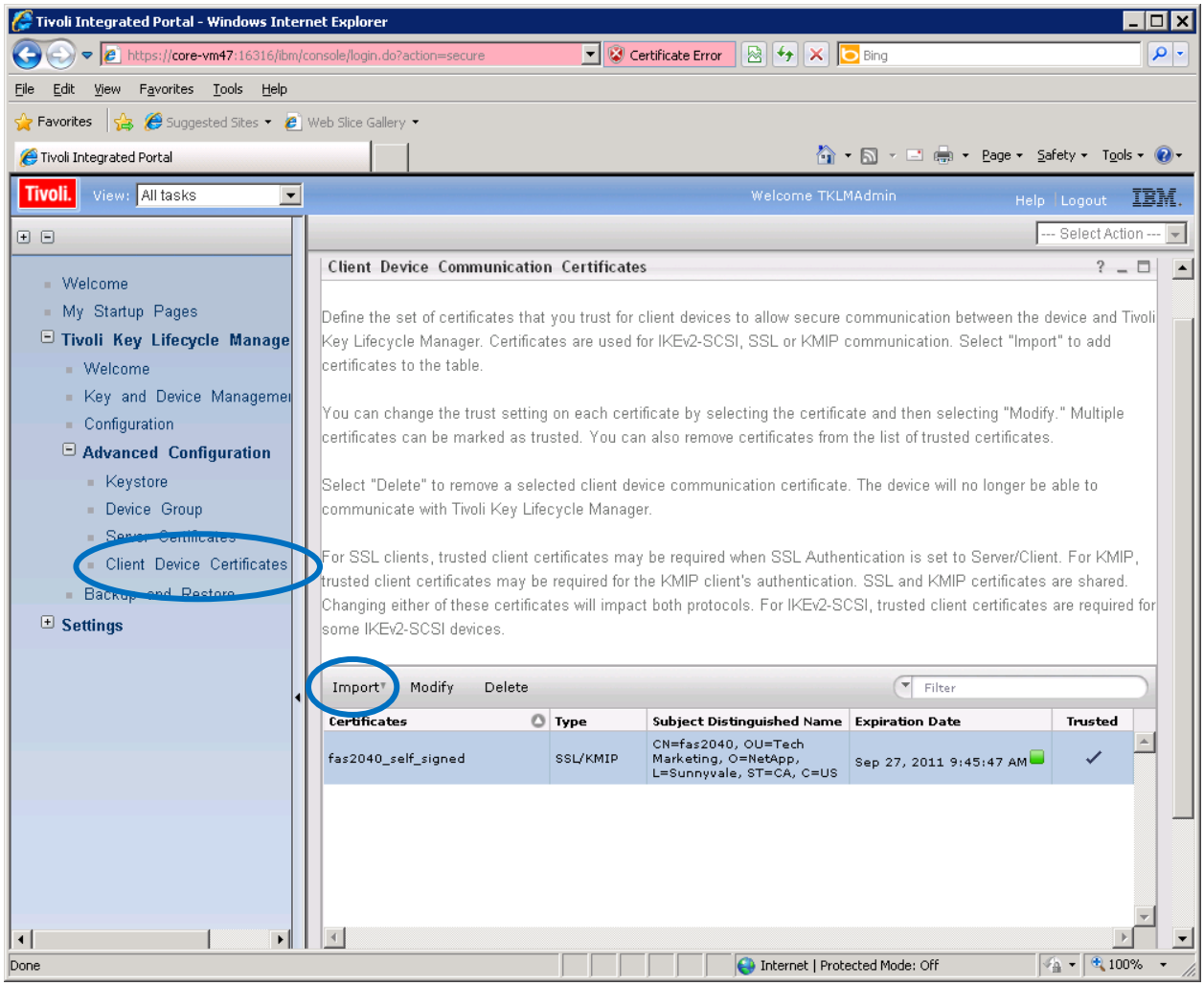
```
core-vm30:~ # openssl x509 -req -days 365 -in client.csr -signkey
client_private.key -out client.pem

Signature ok

subject=/C=Your Country/ST=Your State/L=Your City/O=Your Company/OU=Your
OU/CN=fas2040c-
svl04.iops.eng.netapp.com/emailAddress=your_name@your_company.com

Getting Private key

core-vm30:~ #
```

Note that the CSR is signed using the client_private.key generated earlier and the output file is client.pem.

C.4 IMPORT THE SELF-SIGNED NSE CERTIFICATE INTO TLKM

Instead of importing the CA certificate here, you would import the self-signed client.pem file for each NSE client generated as generated earlier.

Figure 53: Importing Self-Signed NSE Certificate into TKLM



C.5 HA CLUSTER CONSIDERATIONS FOR SELF-SIGNED CERTIFICATES

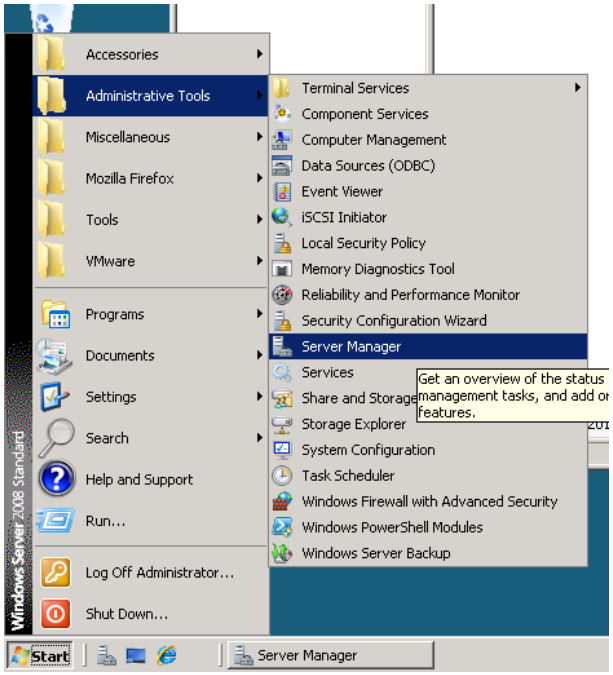
For an HA system such as the FAS2040A, instead of creating individual self-signed certificates for each controller, you would simply take the client.pem, client_private.pem, and tklm_self_signed.pem files used for the first controller and copy the exact same files onto the second controller. This will allow both controllers to fail over properly and take ownership of each other's disks.

APPENDIX D: INSTALLING WINDOWS 2008 CERTIFICATE AUTHORITY SERVICES

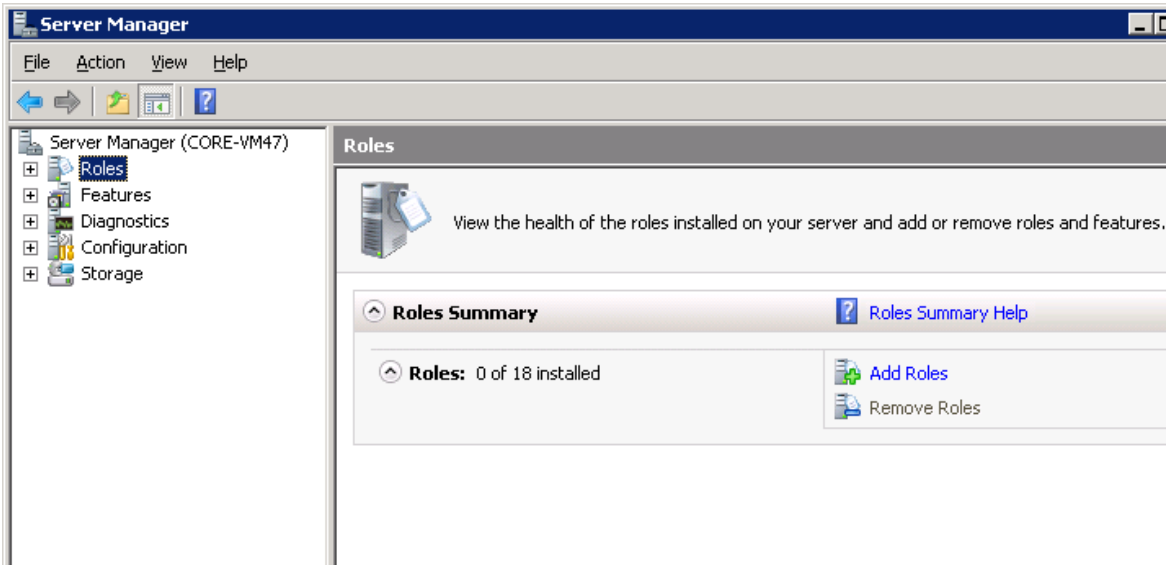
Complete instructions for installing and configuring a certificate authority (CA) server in Windows 2008 can be found on the Microsoft® Web site. The following is a summary of what you need to configure a basic setting. All default values are selected.

Begin by selecting Server Manager from the Windows 2008 Start menu.

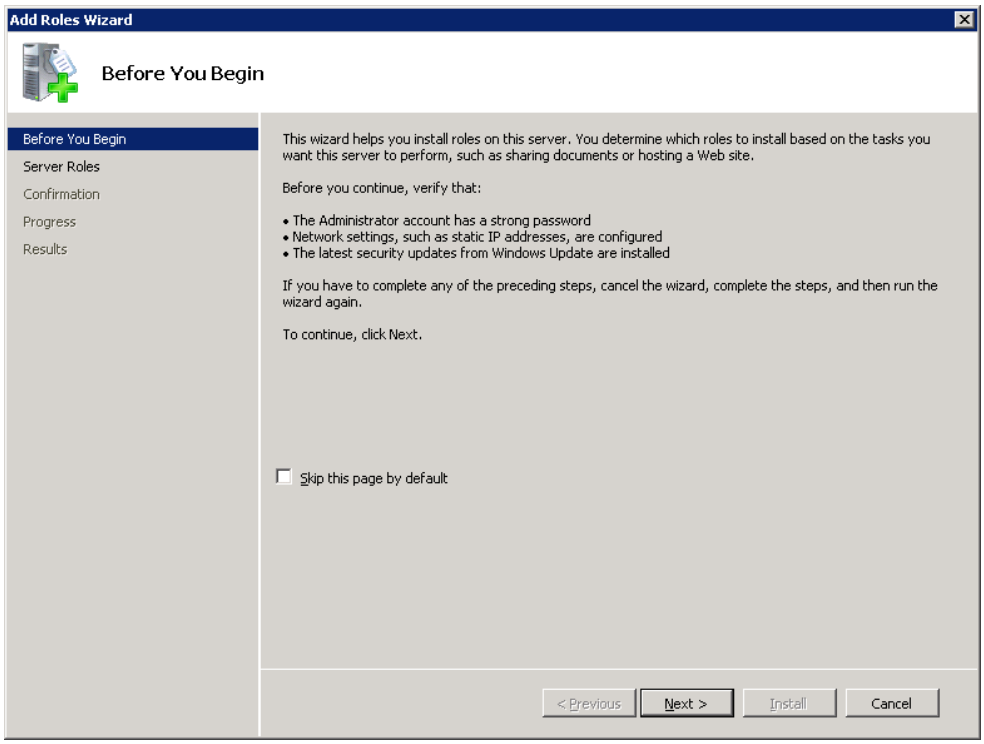
Figures 54-55: Adding CA services in Windows 2008



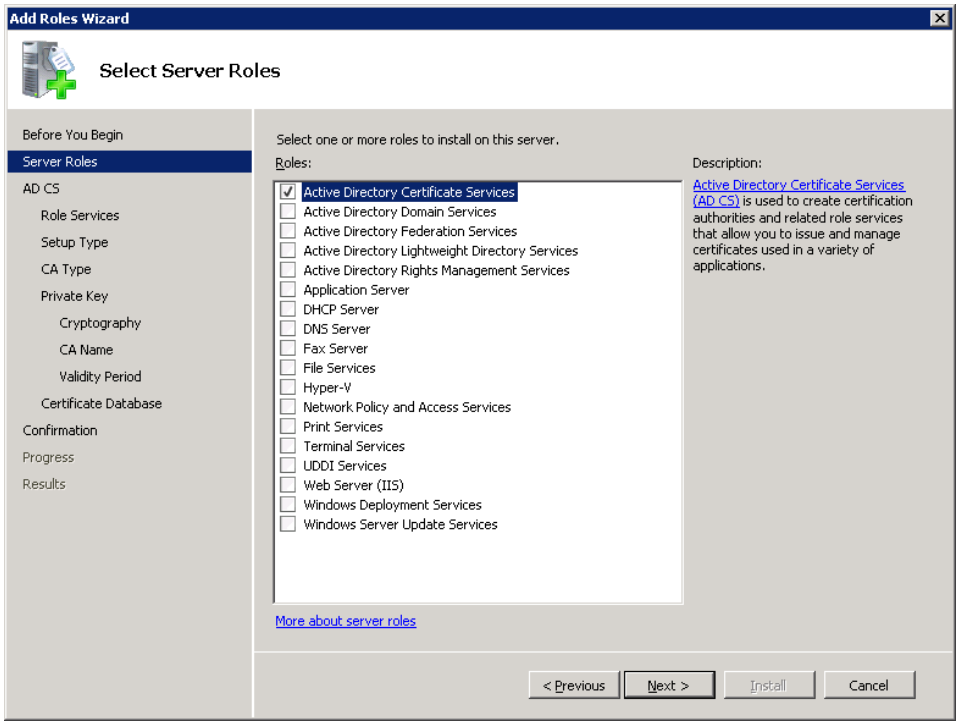
Click Add Roles to add the CA Services role.



Figures 56-57: Configuring the CA Services in Windows 2008

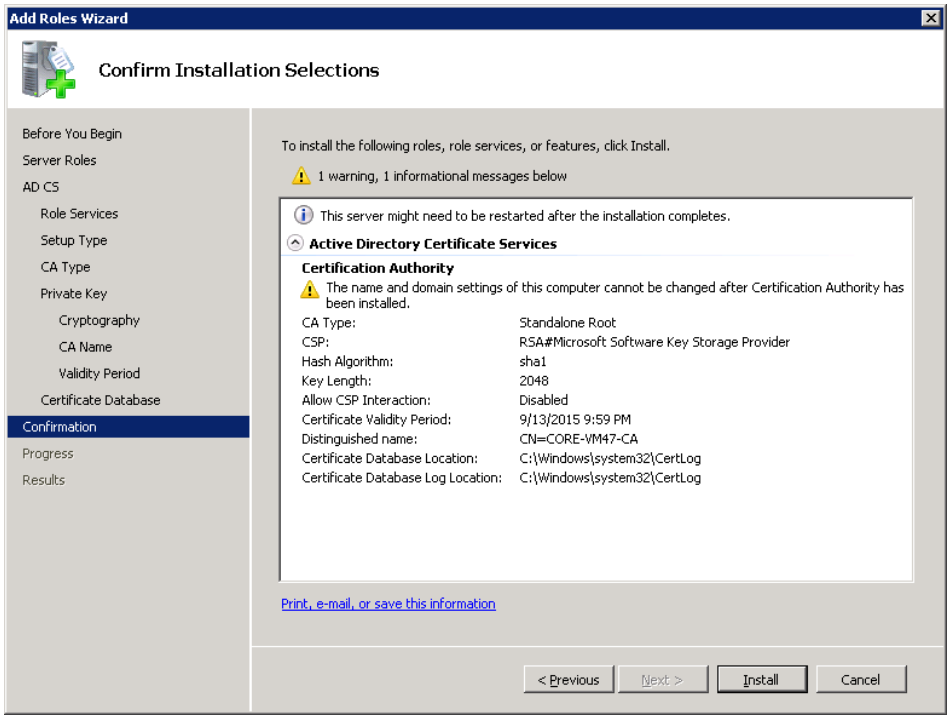


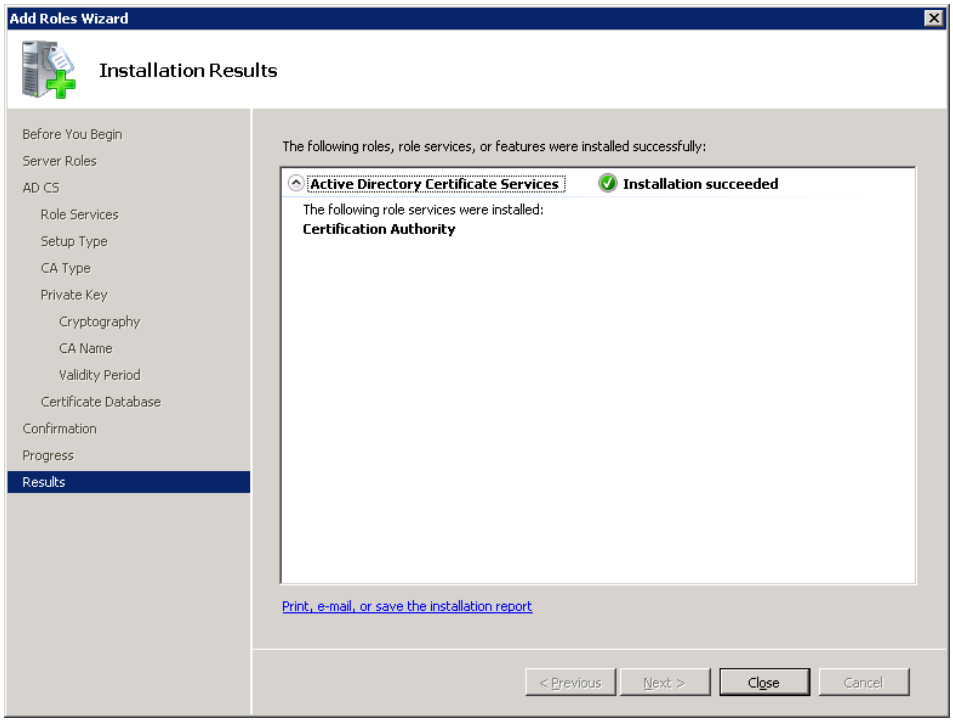
Select the Active Directory Certificate Services to install the CA services. You can optionally install the Web Server (IIS) services here, but this step is optional if you want to enable Web management of the CA server.



The remaining pages are not shown, but all defaults should be selected to get the following confirmation page.

Figures 58-59: Confirm CA Services Installation in Windows 2008



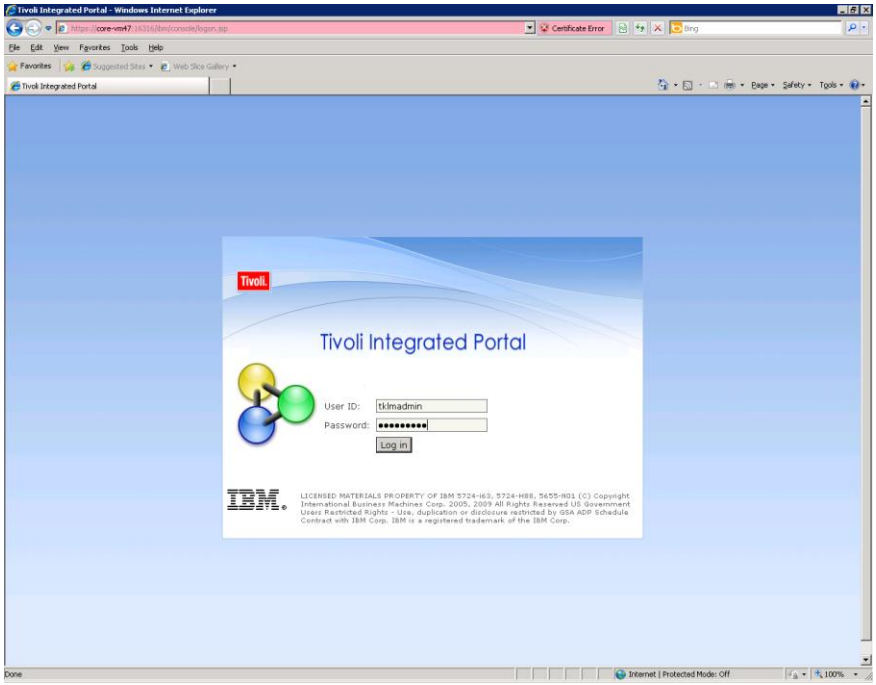


You have now successfully installed a Windows 2008 CA server.

APPENDIX E: CONFIGURATION OF IBM TKLM SERVER FOR WINDOWS

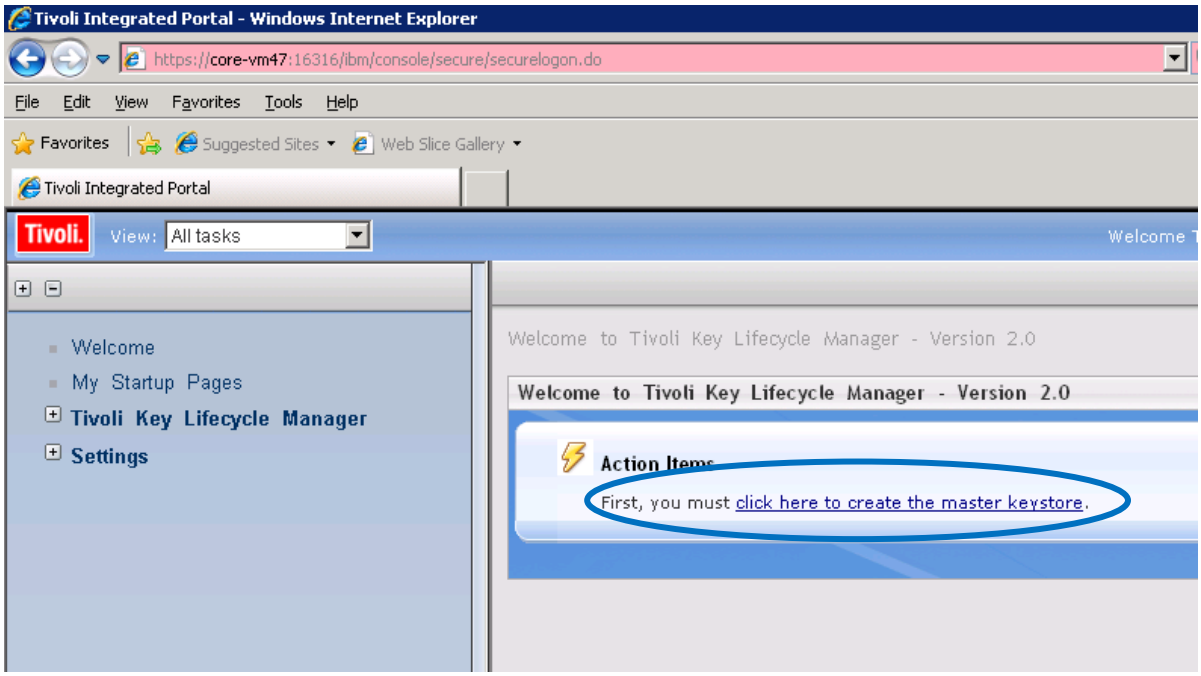
Install the Windows TKLM server using default settings and log into the main portal.

Figure 60: IBM TKLM Login Portal



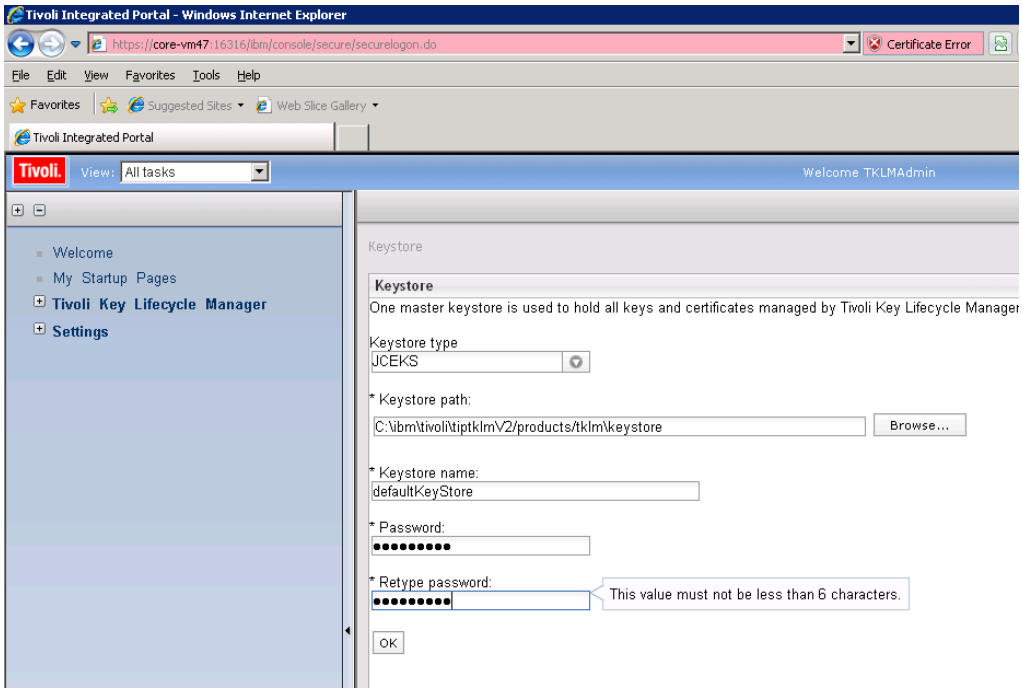
E.1 CREATE THE MASTER KEYSTORE

Figure 61: Creating the Master Keystore

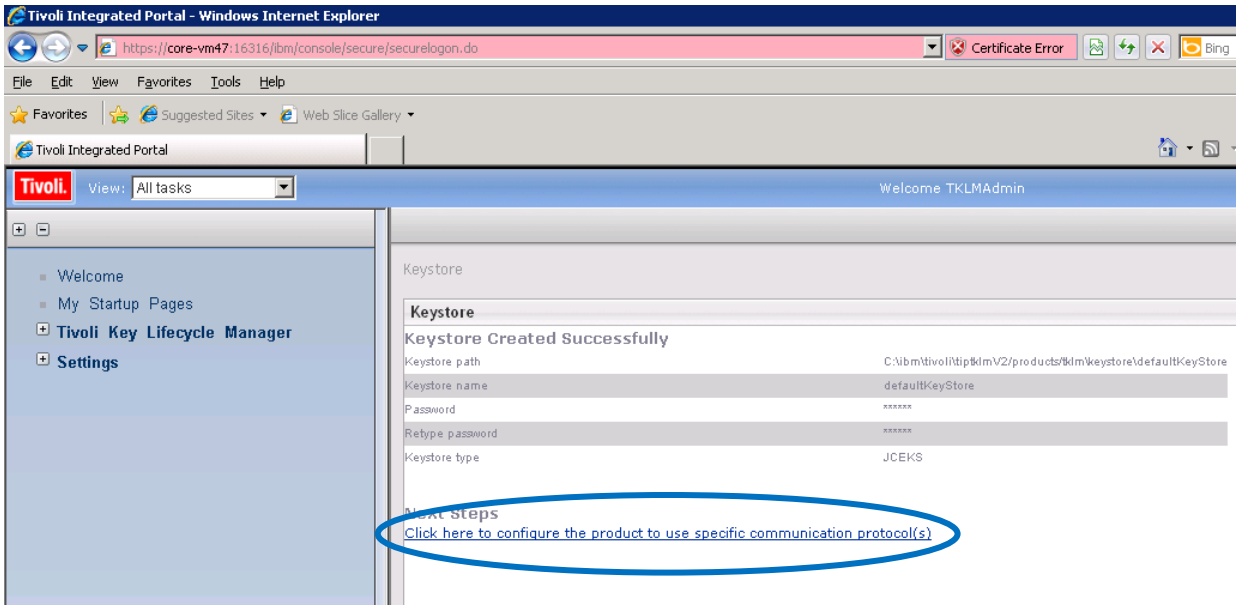


Leave the default keystore name and enter a password for the master keystore.

Figure 62-63: Creation of Master Keystore



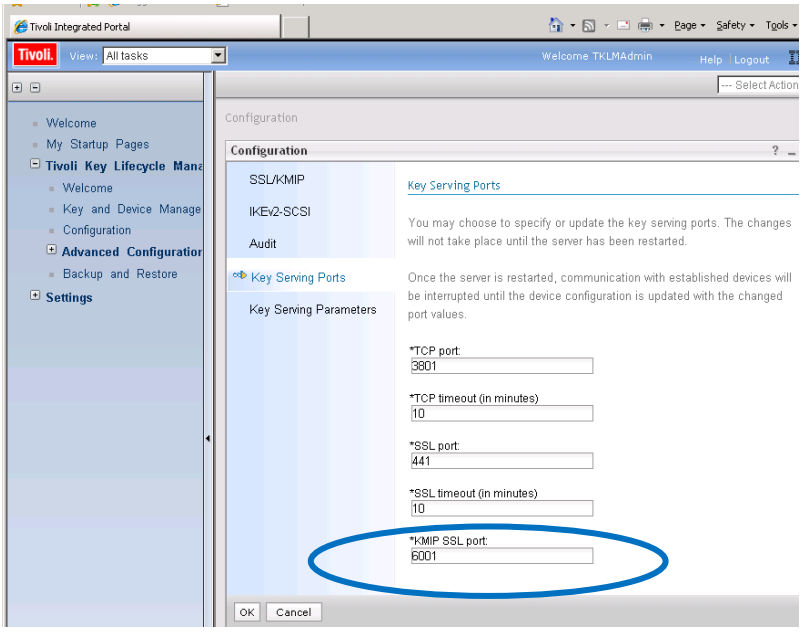
Finish configuring the TKLM server. The next step tells you to configure the communication ports, but it actually will take you to the TKLM server certificate generation page.



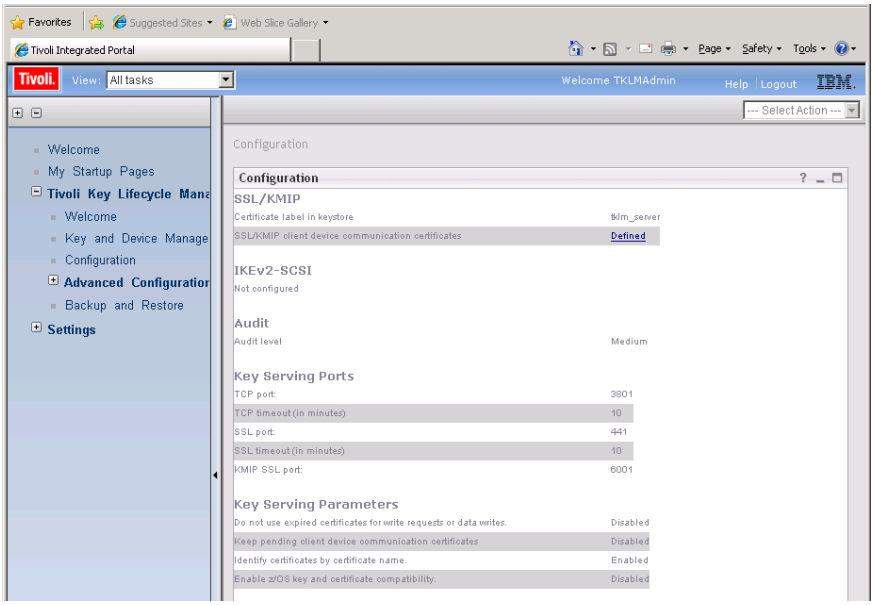
E.2 CONFIGURE THE COMMUNICATION PORTS FOR NSE

All ports should be left at defaults. In this example, the KMIP SSL port has been changed to 6001, but the default port of 5696 can be used as long as this value is set during “key_manager setup” during the NSE setup process. Refer to the document “Data ONTAP 8.1 7-Mode Software Setup Guide” and locate the section on storage encryption for further guidance.

Figure 64-65: Configuring Communication Ports for TKLM

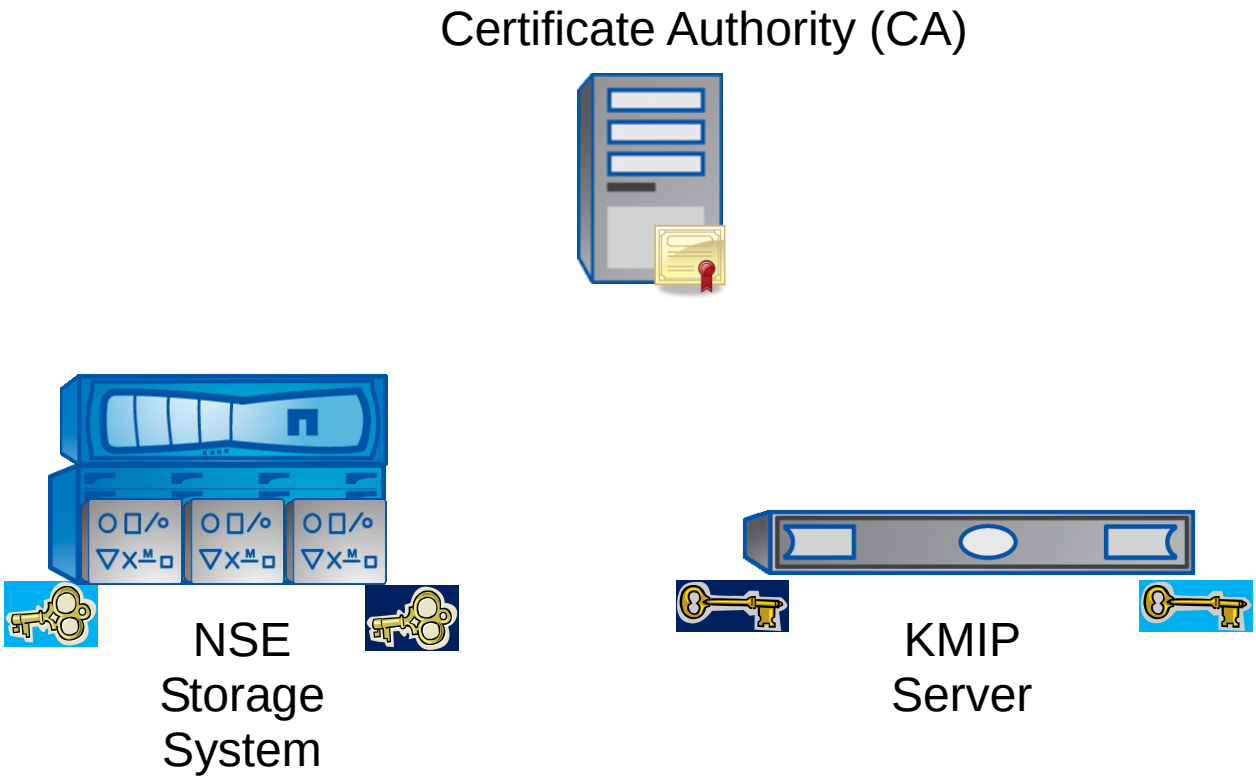


Confirm all configured values are correct.



APPENDIX F: CERTIFICATES 101

Figure 1) Key generation and certificate authority.



A key exchange needs to occur between two entities, and a CA helps ensure trust between the key exchanges. Each party creates a public (dark blue) key and a private (light blue) key to establish a secure session for communication. The public keys are sent to the CA for signing and then exchanged between the two parties. The CA provides a root of trust to make sure the corresponding public keys are valid and haven't been tampered with.

In this guide, the following keys/certificates are created:

- CA public key: used to verify the exchanged signed keys from NSE and the KMIP server
- Public and private keys for both NSE and the KMIP server
- Signed public key certificates for both NSE and the KMIP server

Go further, faster®

www.netapp.com