Technical Report

NetApp Storage Encryption: Preinstallation Requirements and Procedures for SafeNet KeySecure

Mike Wong, NetApp
Neil Shah, NetApp
April 2013 | TR-4074

|Version 1.2|

NetApp Storage Encryption Preinstallation Steps

NetApp® Storage Encryption (NSE) requires that a number of components be installed and configured prior to use with Data ONTAP®. This includes Key Management Interoperability Protocol key management server (KMIP server) configuration, Secure Sockets Layer (SSL) certificate creation and signing and installation, and manual entry of Data ONTAP bootloader variables. This guide offers a step-by-step example of the preinstallation steps using the SafeNet KeySecure k460 appliance as the KMIP-compatible server as well as the certificate authority (CA) and OpenSSL for Linux® certificate generation.

TABLE OF CONTENTS

1 Introduction 3

2 Required Bootloader Variables for NSE 3

2.1 Confirm Version of Data ONTAP 3

2.2 Configure Bootloader Environment Variables 3

3 SSL Certificate Creation 4

3.1 Creating Certificate Signing Requests (.CSR Files) 5

3.2 Send CSR Files to CA to Sign and Request CA Public Certificates 8

3.2.1 Local CA Server Configuration and Signing CSR files by Local CA Server 9

3.2.1.1 KeySecure Local CA Server Configuration 9

3.2.1.2 Signing CSR files by KeySecure Local CA Server 15

3.2.2 External CA Server Configuration and Signing CSR files by External CA Server 20

3.2.2.1. External CA Server Configuration 20

3.2.2.2. Signing CSR files by External CA server 27

3.3 Create PEM Files for NSE 27

3.4 Copy Keys and Certificates to NSE 28

3.5 Create KMIP Cryptographic Key Server Configuration 28

4 Verification of PEM Files 29

5 HA Cluster Pair SSL Certificate Considerations 32

6 KeySecure HA Cluster Pair Considerations 32

APPENDIX 33

Certificate Cleanup 33

Removal of Existing Certificates 33

SSL Certificate Replacement 33

Enabling FIPS Compliance Mode on KeySecure 34

Enable FIPS Compliance 34

Creation of User Account 36

Configure Authentication Object in Cryptographic Key Server 37

Certificates 101 40

1 Introduction

NetApp Storage Encryption has a number of preinstallation steps that must be completed before use with 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 variables that must be configured 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 to reach the key servers during boot. Because of its use during boot, it cannot participate in network trunking or VIF configuration. After boot completes, Data ONTAP may use it as a standard single interface.
- **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

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 CA to sign the certificates. An example using self-signed certificates can be found in the appendix. The following SSL certificates will be generated:

- 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.
- CA public certificate
  - This is an exported public certificate from the CA.
  - If there is a CA chain, all CA certificates must be concatenated together.
  - This file needs to be renamed <IP_Address_of_KMIP_Server>_CA.pem for NSE use.

3.1 Creating Certificate Signing Requests (.CSR Files)

The first step in gathering the necessary certificates is to generate the .CSR files from both KeySecure and NSE.

Create KeySecure .CSR File

The following steps assume the KeySecure has completed the initial configuration with the iKeys and PED devices. Consult your KeySecure Installation Guide for steps on how to complete the initial configuration of the KeySecure.

Figure 1) Creation of KeySecure KMIP SSL certificate.

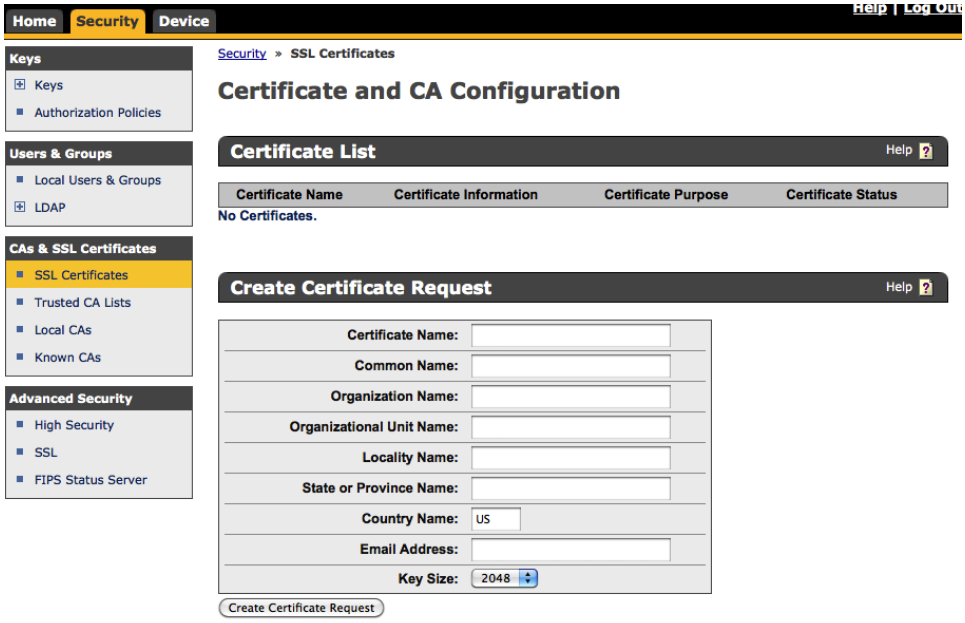


Figure 2) Create KeySecure KMIP SSL certificate.

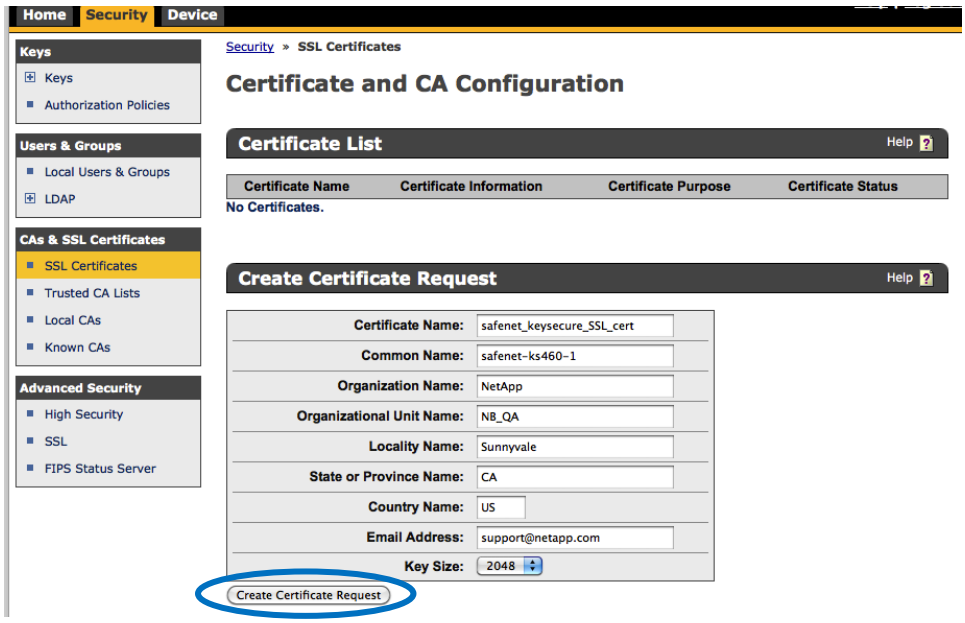


Figure 3) KeySecure CSR needing to be signed.

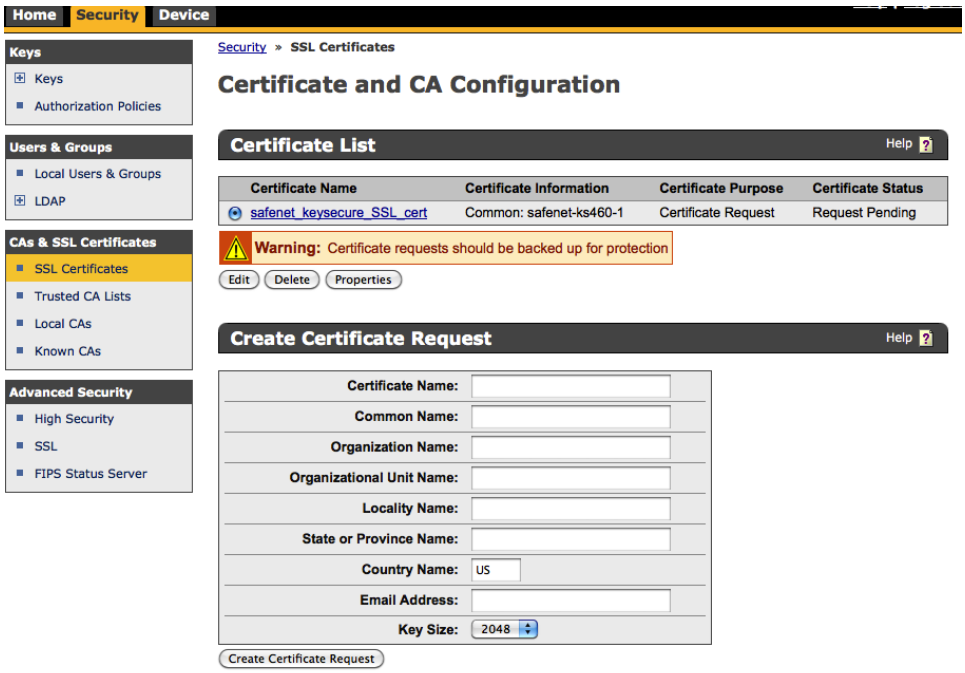


Figure 4) Download CSR.

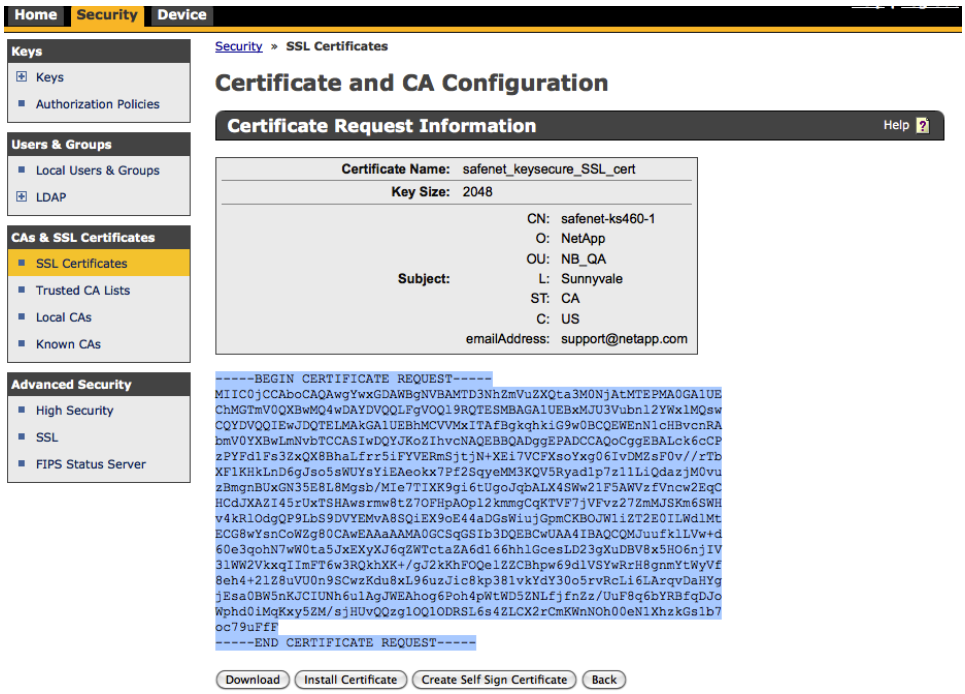
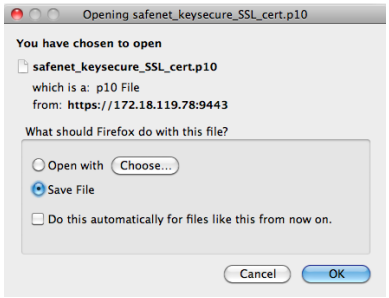


Figure 5) Save CSR file.



The resulting file (safenet_keysecure_SSL-cert.p10) is a PKCS#10 certificate signing request, which needs to be saved and sent to the CA for signing.

Create NSE .CSR File

This step needs to be done external to the NSE system. A public and private key pair can be generated in either Windows or any UNIX variety 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) []:hostname of YOUR NSE system
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
```

3.2 Send CSR Files to CA to Sign and Request CA Public Certificates

You now have two CSR files:

- safenet_keysecure_SSL_cert.p10 – SafeNet KeySecure CSR
- client.csr – NetApp Storage Encryption CSR

These files need to be signed by CA. The CA can be one of the following

1. The local CA server on the KeySecure.

2. A 3rd party single external CA server or multiple external CA servers (hierarchical)

Send these files to the CA for signing and at the same time request the public certificates from all CA servers involved in the CA hierarchy (if any). When the CSR files are signed by the CA servers, the resulting file needs to be in base 64 encoded X.509 format.

One of the more common CA hierarchies involves an external intermediate CA and a an external root CA hierarchy.

If using the KeySecure local CA server, read section 3.2.1

If using a 3rd party external CA server or external CA server hierarchy, read section 3.2.2

3.2.1 Local CA Server Configuration and Signing CSR files by Local CA Server

At this point we have created two CSR files – KeySecure CSR and NSE CSR. These files need to be signed by the KeySecure k460's Local CA. In this section we will configure the Local CA server and then have the Local CA sign the KeySecure and NSE CSR files

3.2.1.1 KeySecure Local CA Server Configuration

Creating Local CA on KeySecure

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. A customer may choose to use their own CA and export the file for you. The following example shows creation of a CA in KeySecure

Figure 6) Creation of self-signed root CA certificate under local CAs.

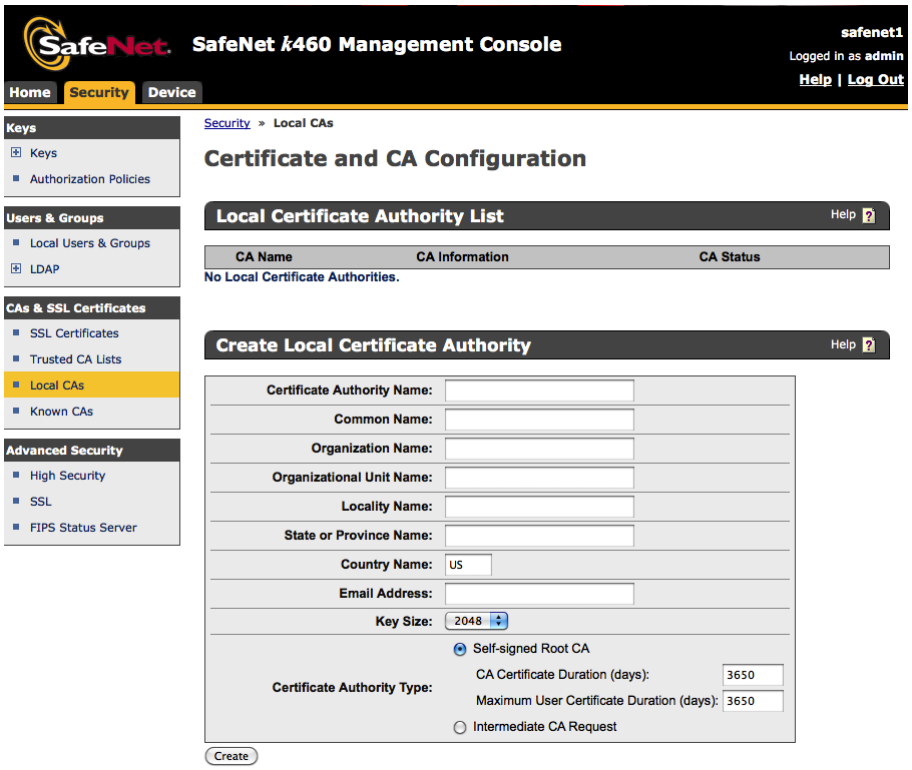


Figure 7) Specify CA certificate parameters.

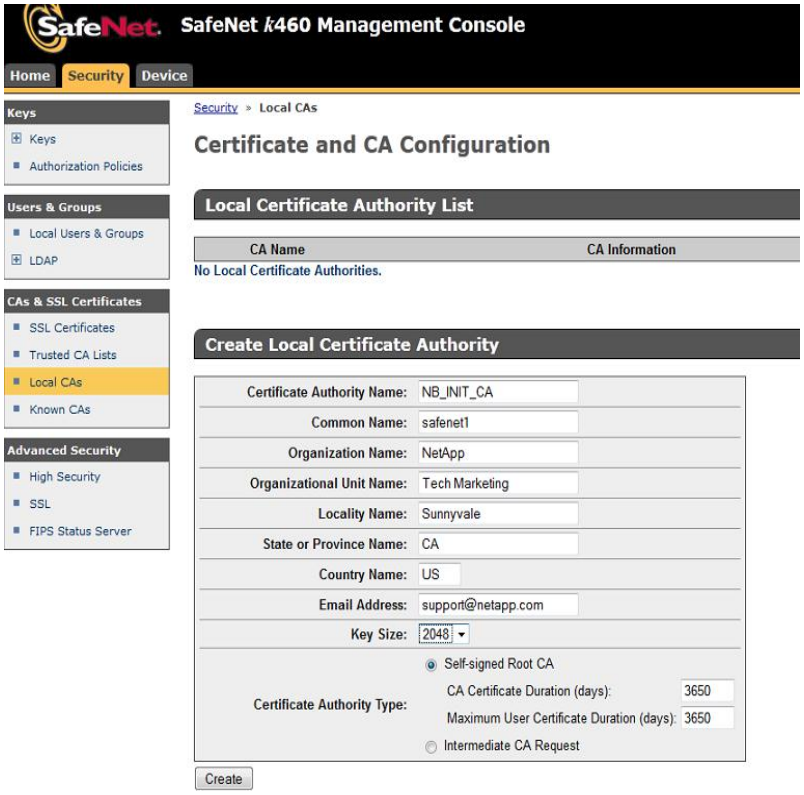


Figure 8) Root CA certificate created.

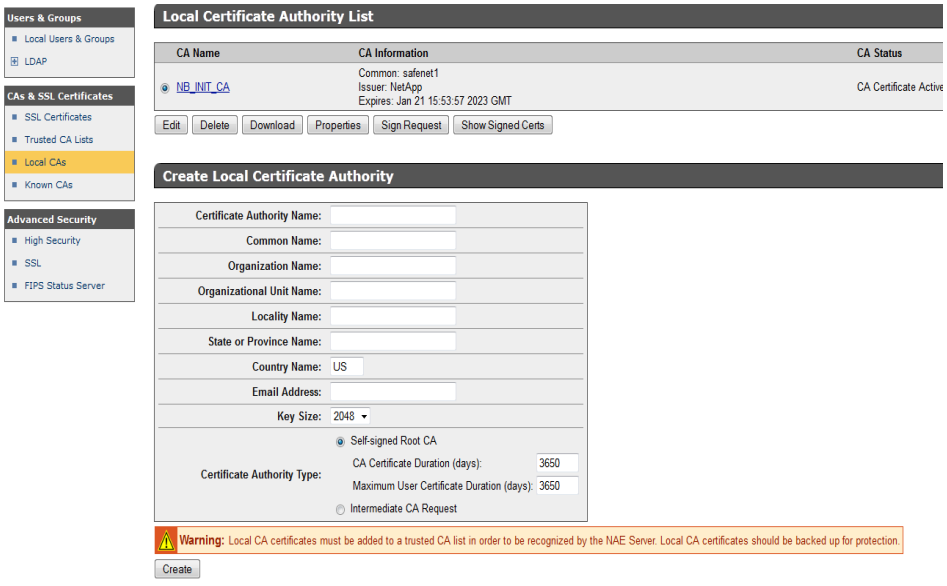


Figure 9) Download CA public certificate.

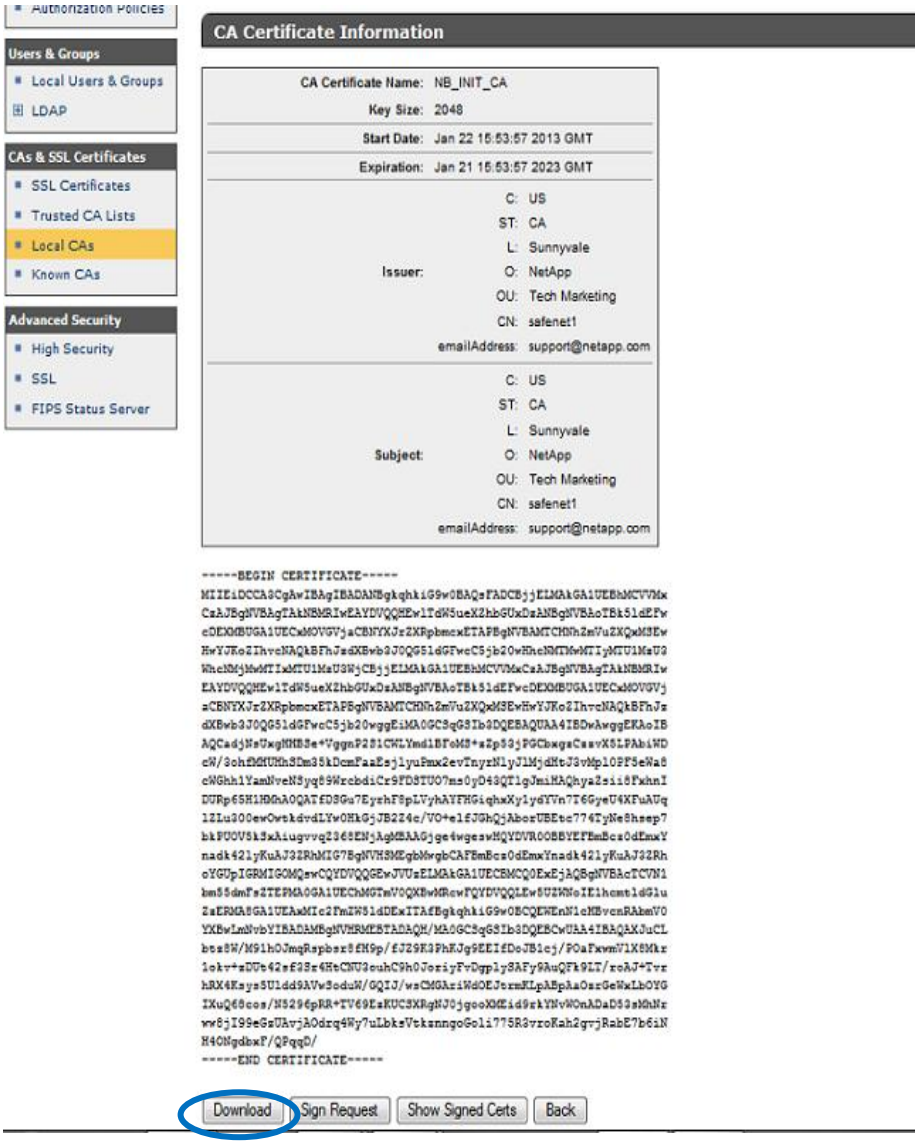
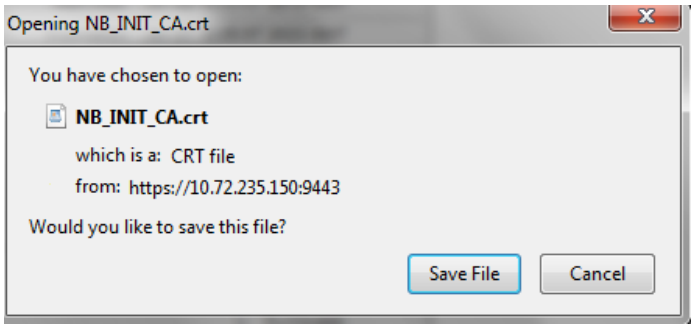


Figure 10) Save root CA public certificate.



Alternatively you can also select the text from ----BEGIN CERTIFICATE…END CERTIFICATE---- and copy/paste it into a notepad/wordpad/text file and save it

The CA certificate needs to be named <IP_Address_of_KMIP_Server>_CA.pem. If you have multiple KMIP servers who's certificates will be signed by this CA, 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.

At this point we have successfully created the Local CA server and generated the <IP_Address_of_KMIP_Server_1>_CA.pem file

Adding Local CA to Trusted CA Lists

This step is required for the local CA to be trusted by the key management server.

Figure 11) Trusted CA authority list profiles.

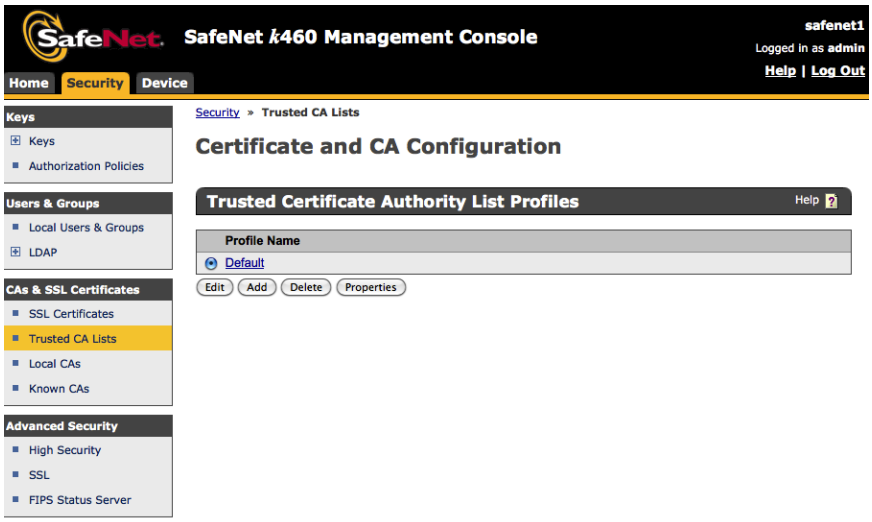


Figure 12) Click on Default

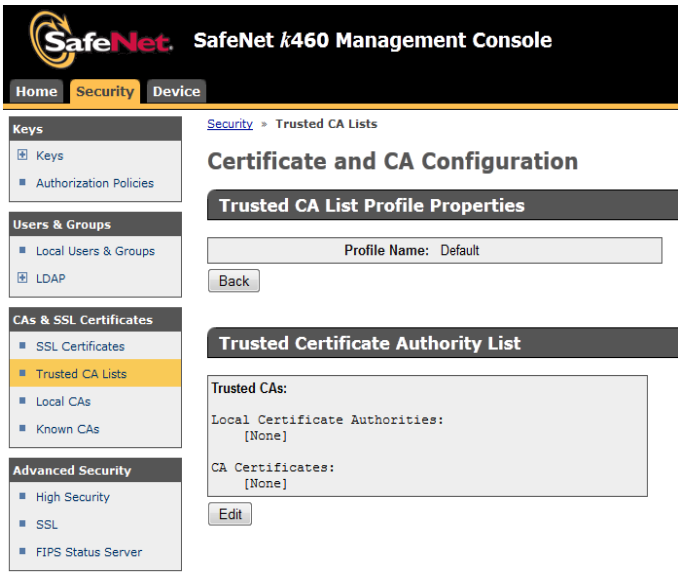


Figure 13) Click Edit and Add root CA to trusted CA list.

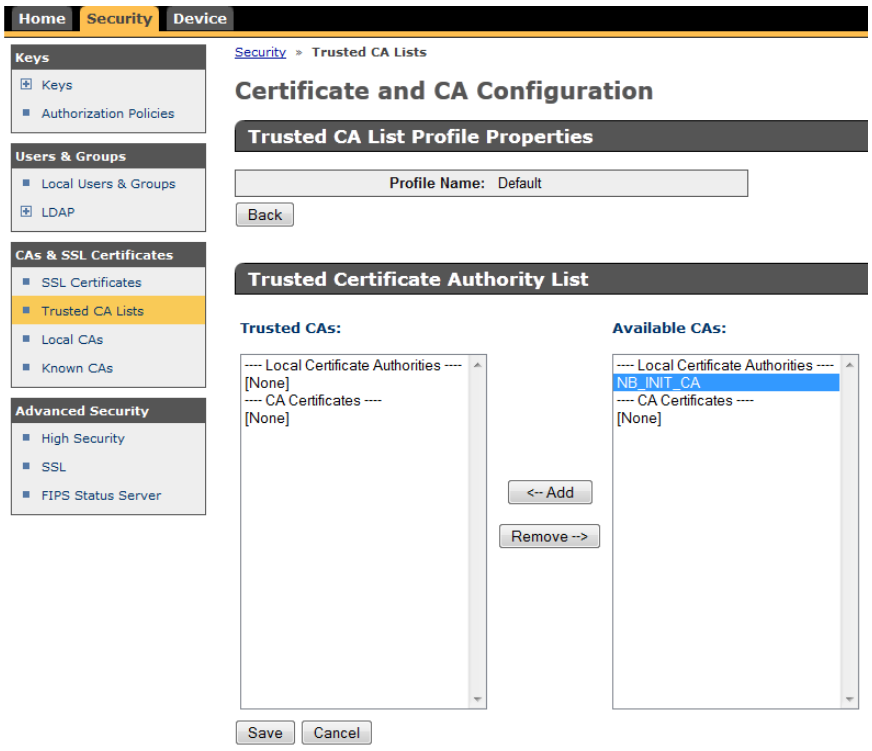


Figure 14) Save modified trusted CA list.

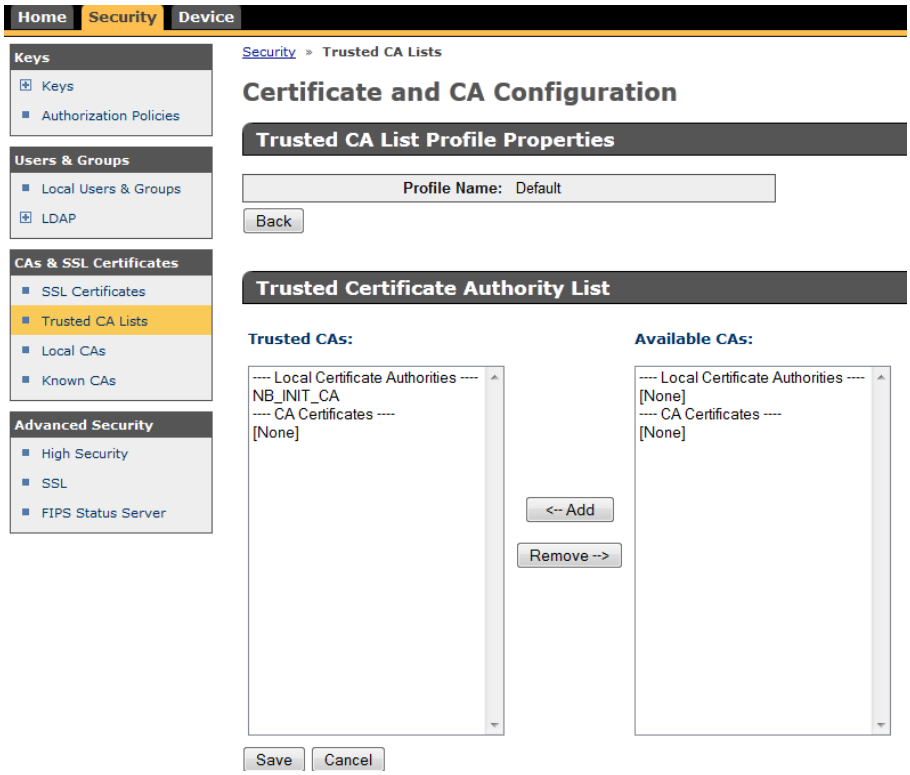
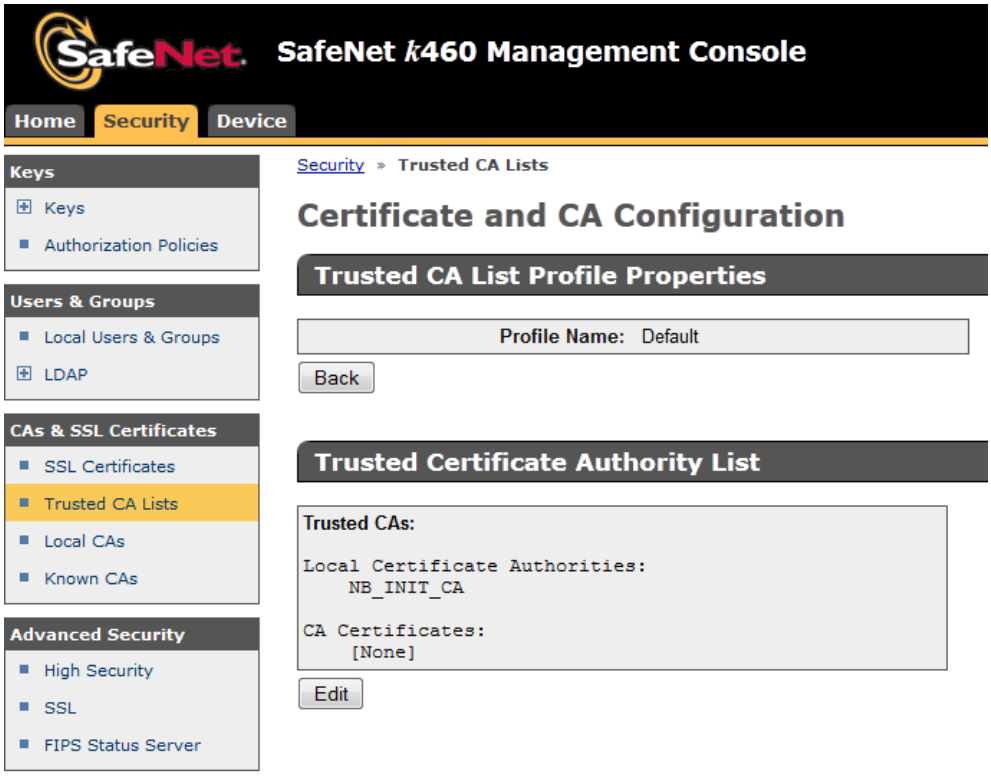


Figure 15) Confirm trusted CA list.



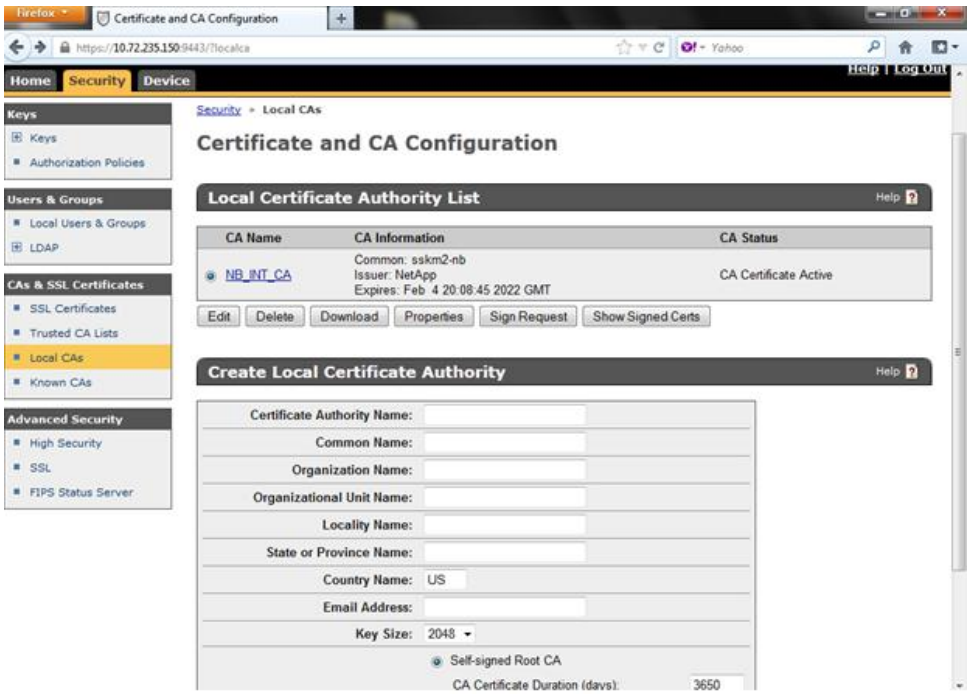
At this point, we have created a Local CA on KeySecure and added that Local CA to the trusted CA list, generated <IP_Address_of_KMIP_Server_1>_CA.pem file. The next step is to sign the CSR files (KeySecure CSR and NSE CSR) by the Local CA

3.2.1.2 Signing CSR files by KeySecure Local CA Server

Remember we have two CSR files (KeySecure CSR and NSE CSR) that need to be signed by the KeySecure Local CA. This section explains the process of signing the CSR files by the Local CA

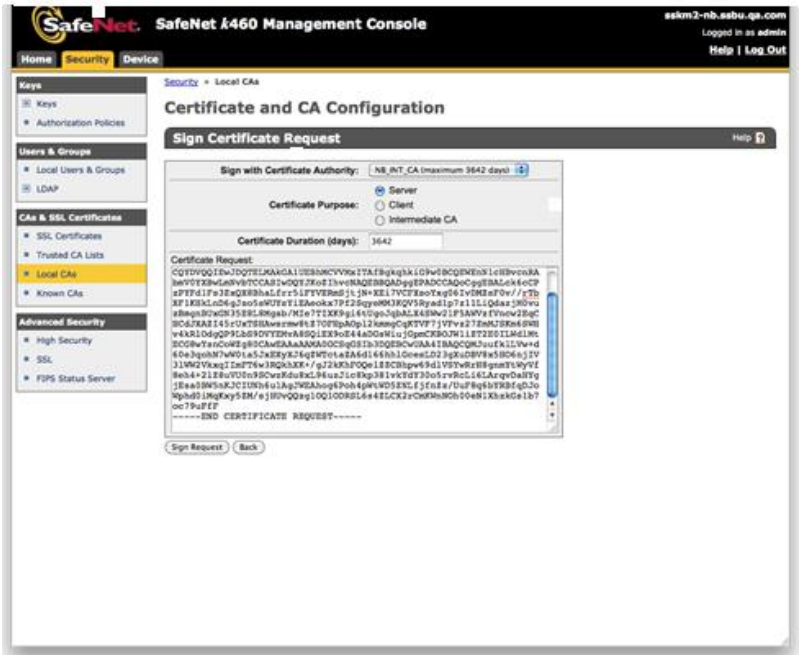
Signing of the KeySecure CSR by the KeySecure Local CA

Figure 16) Select the Local CA. Select Sign Request.



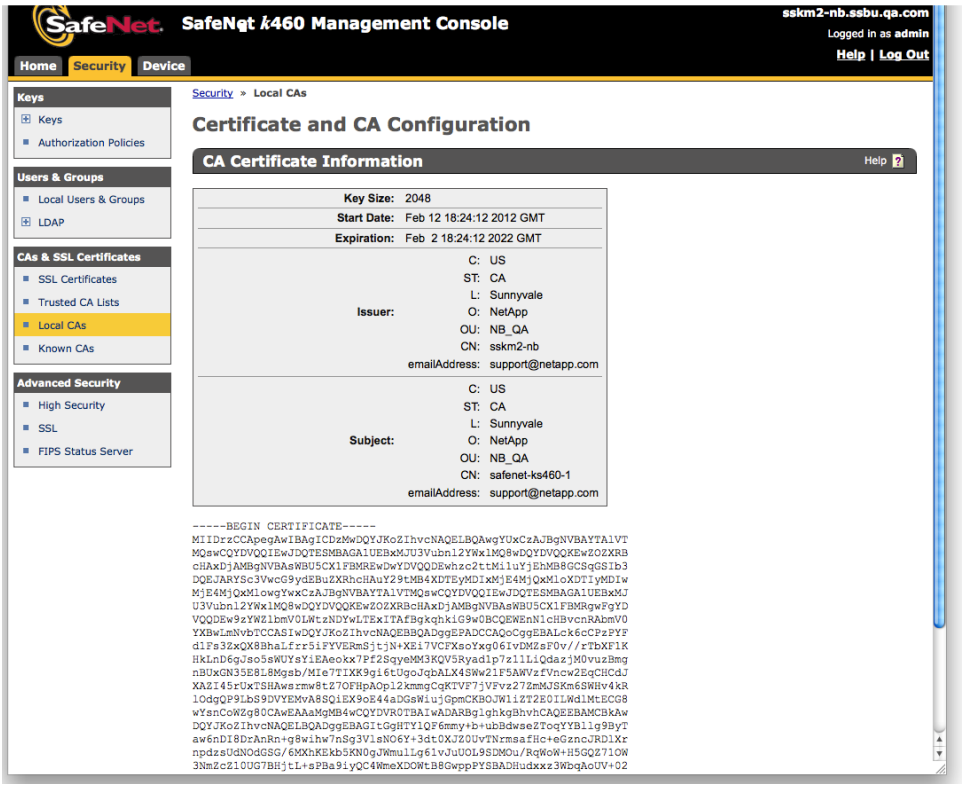
The next step is to sign the KeySecure CSR by the KeySecure Local CA.Open the KeySecure CSR in a notepad/wordpad. Copy the KeySecure CSR contents and paste into the window as shown in Figure 17) below.

Figure 17) Select “Server” and click Sign Request



The next step is to download the signed KeySecure certificate as shown in Figure 18) below in a safe location or alternatively you could also copy and paste the signed KeySecure certificate in a notepad/wordpad/text file and save it

Figure 18) Click “Download.”



Installing Signed KeySecure Certificate into KeySecure

Figure 19) Select the certificate name from the certificate list.

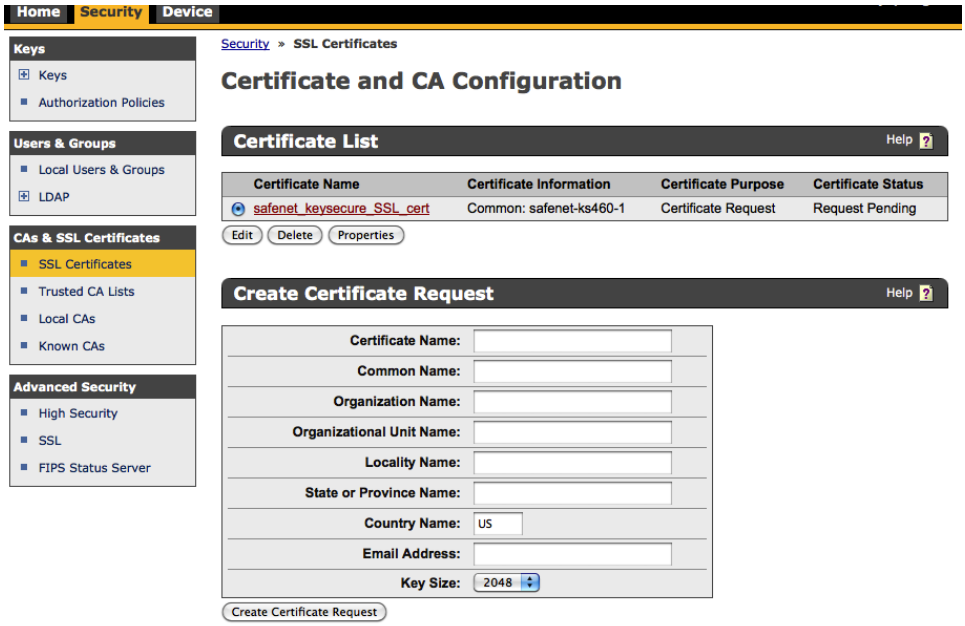
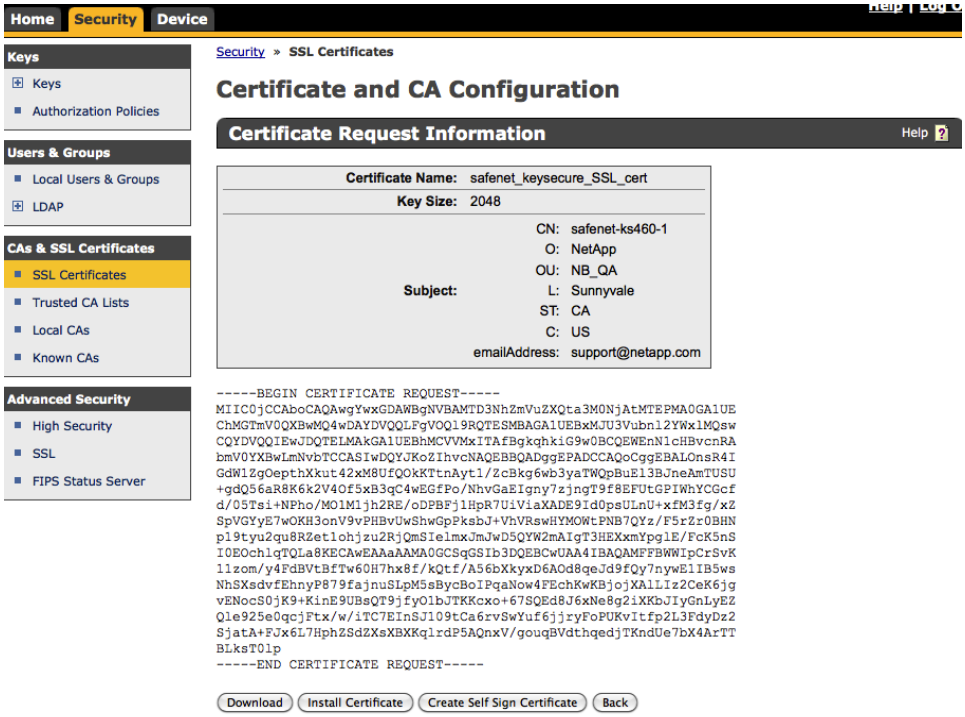


Figure 20) Select install certificate.



The next step is to install the signed KeySecure SSL certificate. Open the signed KeySecure certificate in

a notepad/wordpad. Copy and Paste the contents of the signed KeySecure certificate into the window as shown in Figure 21) below. Then click Save to install the signed KeySecure SSL certificate

Figure 21) Click Save

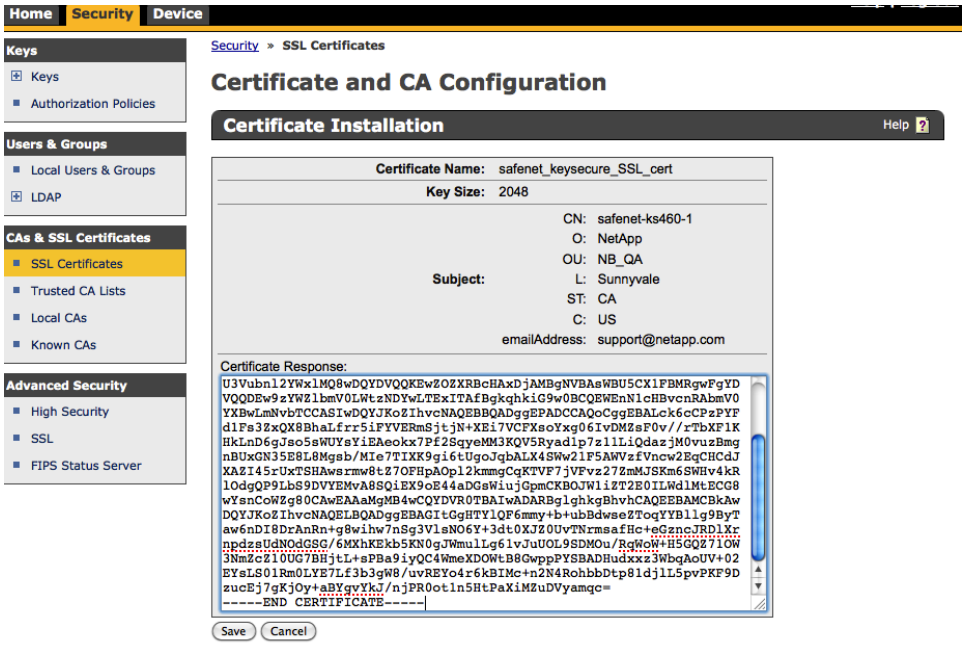
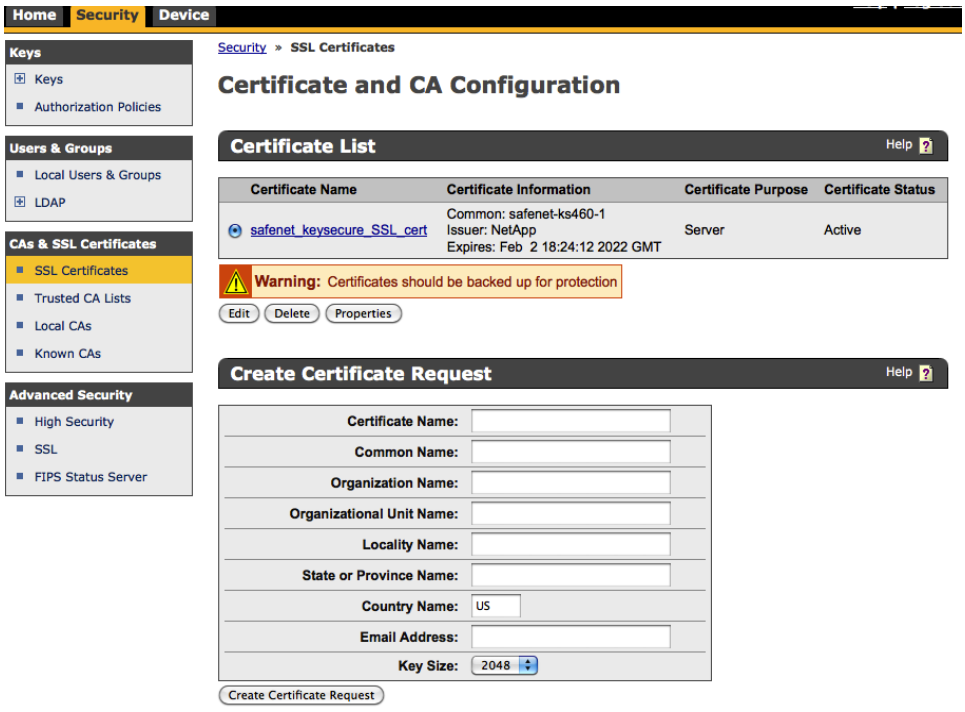


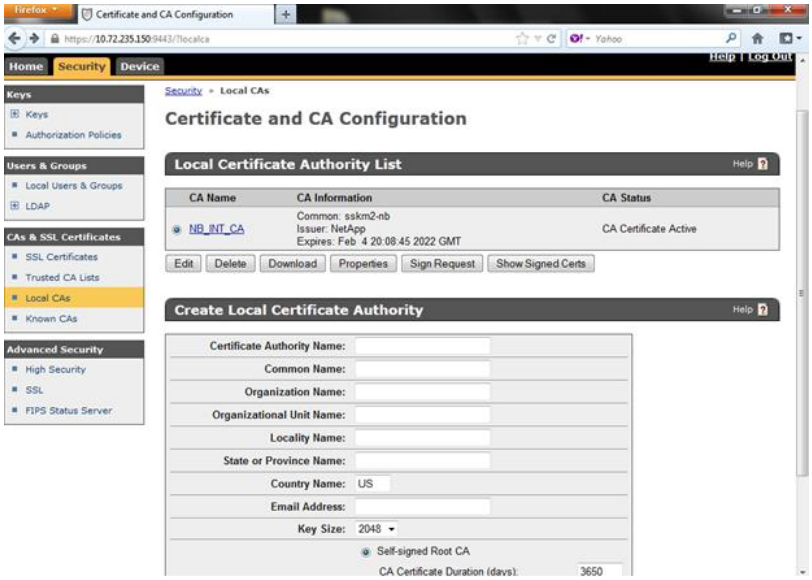
Figure 22) Save signed KeySecure certificate.



At this point, the Local CA has successfully signed the KeySecure CSR and we have installed the signed KeySecure CSR into KeySecure. The next step will be signing the NSE CSR by the Local CA.

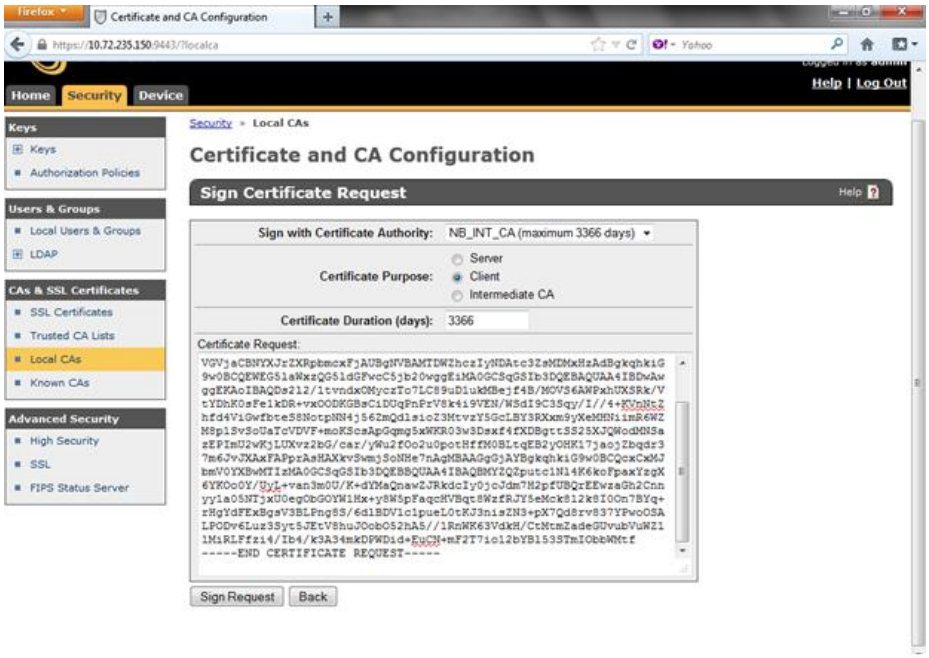
Signing the NSE CSR by the Local CA

Figure 23) Select the Local CA. Select Sign Request.



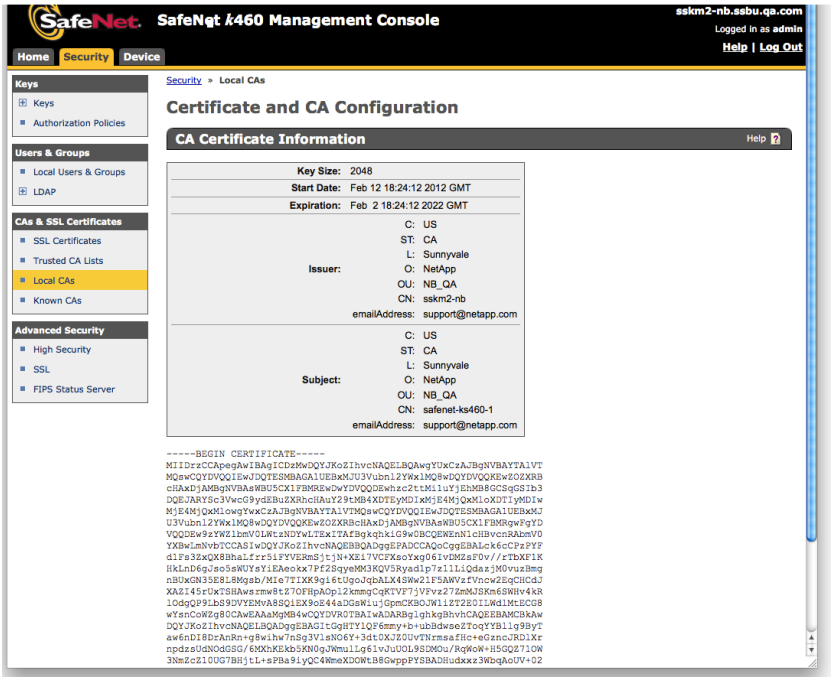
The next step is to sign the NSE CSR by the KeySecure Local CA. Open the NSE CSR in a notepad/wordpad. Copy and Paste the contents of the NSE CSR into the window as shown in Figure 24) below

Figure 24) Select "Client" and click Sign Request.



The next step is to download the signed NSE CSR as shown in Figure 25) below in a safe location or alternatively you could also copy and paste the signed NSE CSR in a notepad/wordpad/text file and save it. This is the client.pem file

Figure 25) Click Download



The Local CA has now also successfully signed the NSE CSR.

At this point, the Local CA has successfully signed both the KeySecure CSR and NSE CSR files. The signed KeySecure SSL certificate has been installed. Save the signed NSE SSL certificate for use in Data ONTAP.

3.2.2 External CA Server Configuration and Signing CSR files by External CA Server

In the previous section, we took you through the steps required for Local CA server configuration and the steps required for the Local CA to sign KeySecure CSR and NSE CSR.

This section will show the steps required for the External CA server configuration and the general guideline that can be used by the External CA to sign the KeySecure CSR and NSE CSR. This section is not required if you are using the KeySecure Local CA to sign your CSR files.

3.2.2.1. External CA Server Configuration

This section describes the steps required for the External CA Server configuration

The example below is based on the following configuration

CA hierarchy containing external root CA server and external intermediate CA server

Import Two External CA Certificates into KeySecure

This step is important so that KeySecure is aware of the CAs used to sign its CSR file.

Figure 26) Import root CA public certificate.

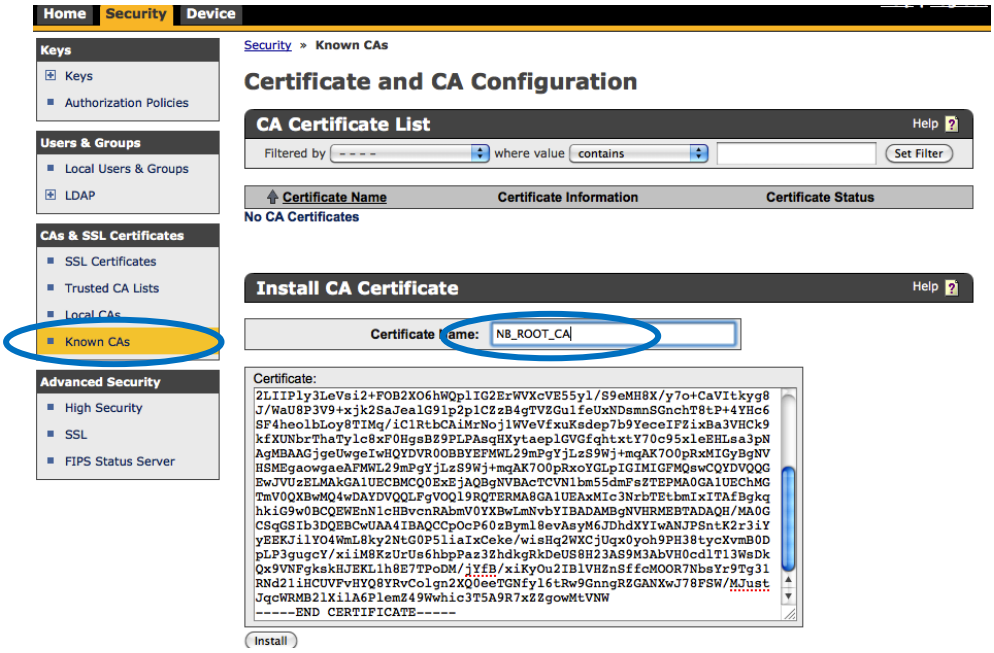


Figure 27) Import intermediate CA public certificate.

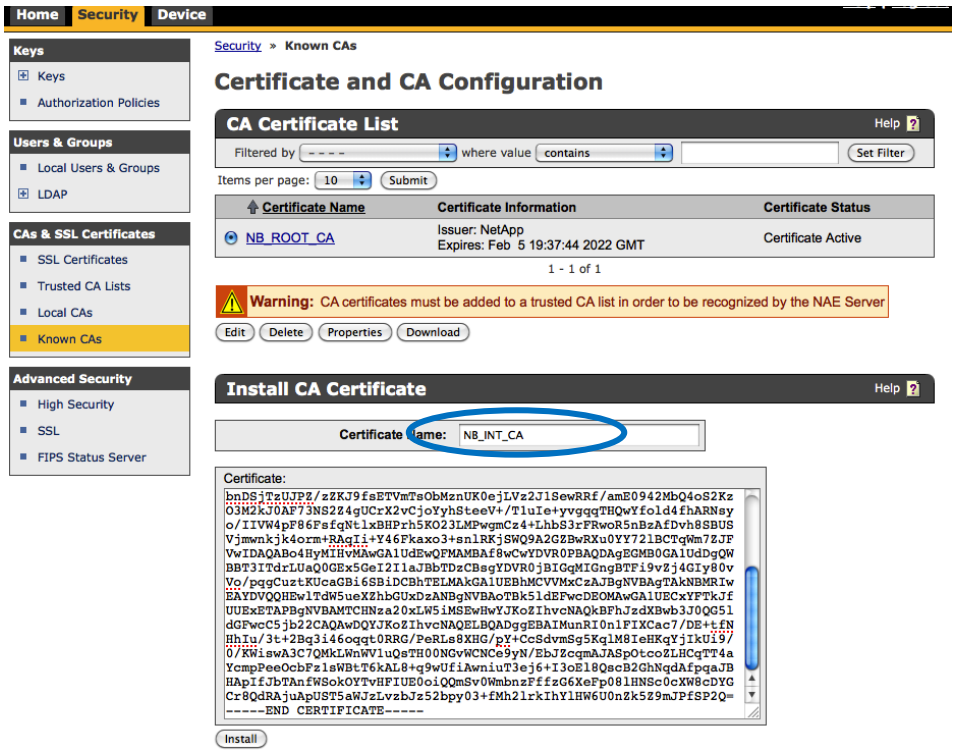
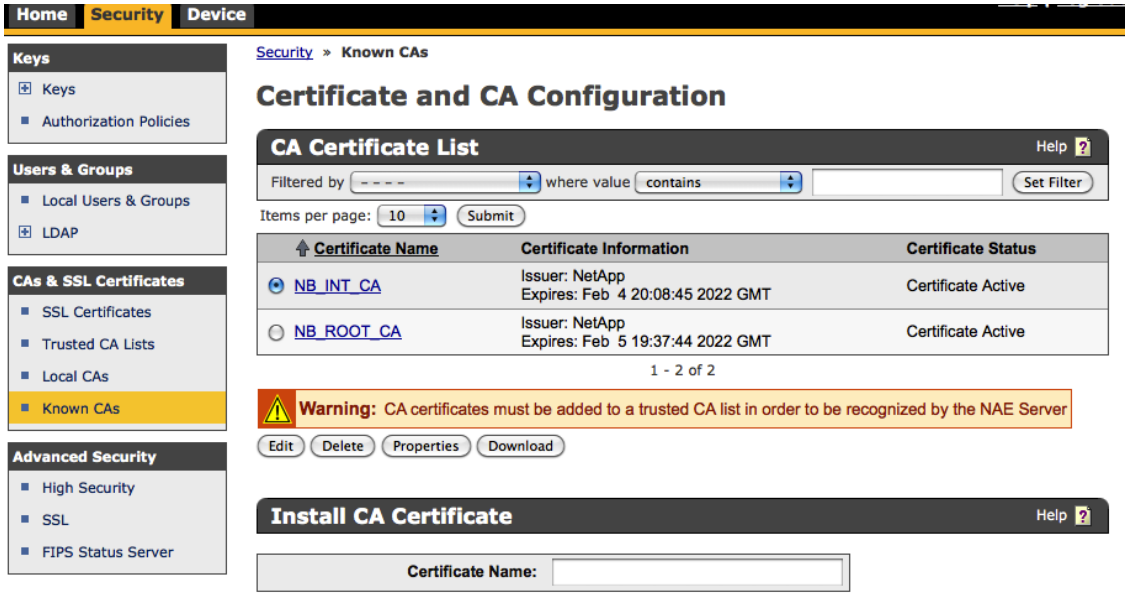


Figure 28) Verify presence of certificate chain.



Add CA Servers to Trusted CA List for KeySecure

This step is required in order for the KeySecure to trust the CAs used to sign its CSR.

Figure 29) Trusted CA list profiles.

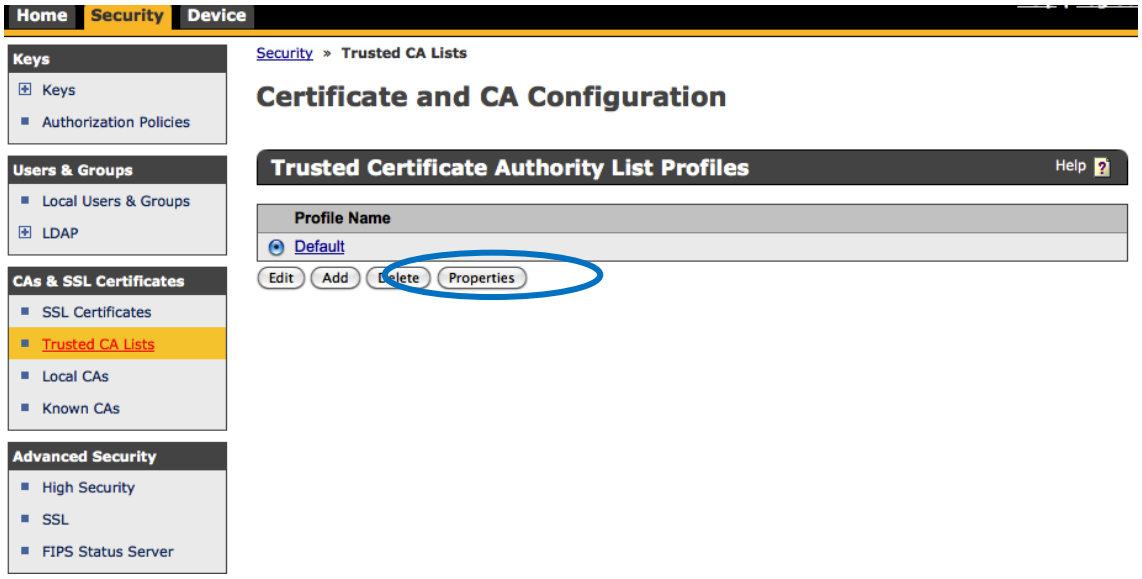


Figure 30) Edit trusted CA list profile.

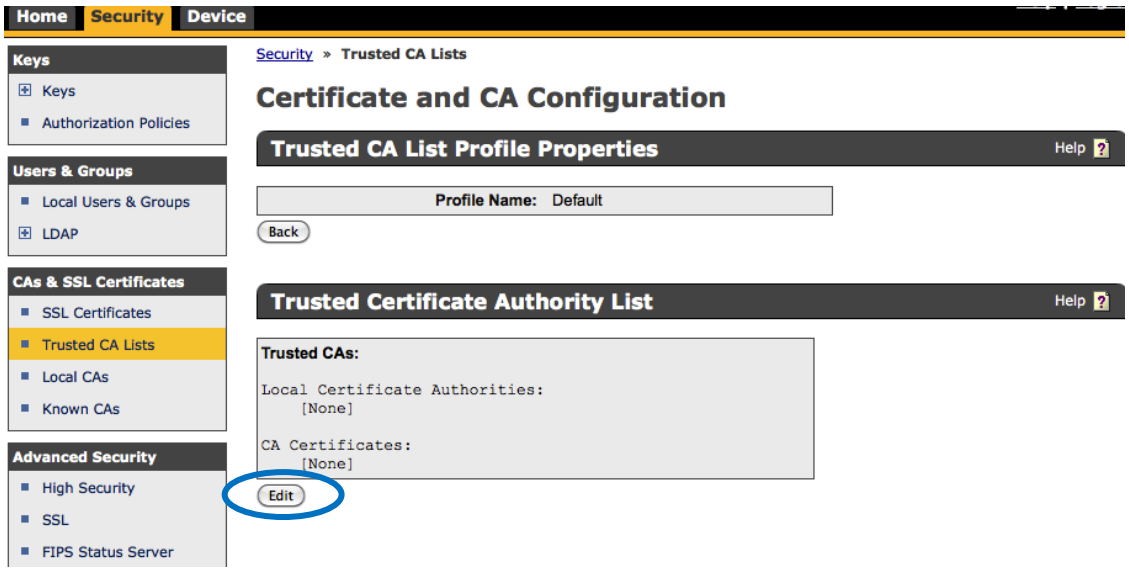


Figure 31) Add CAs to trusted CA list.

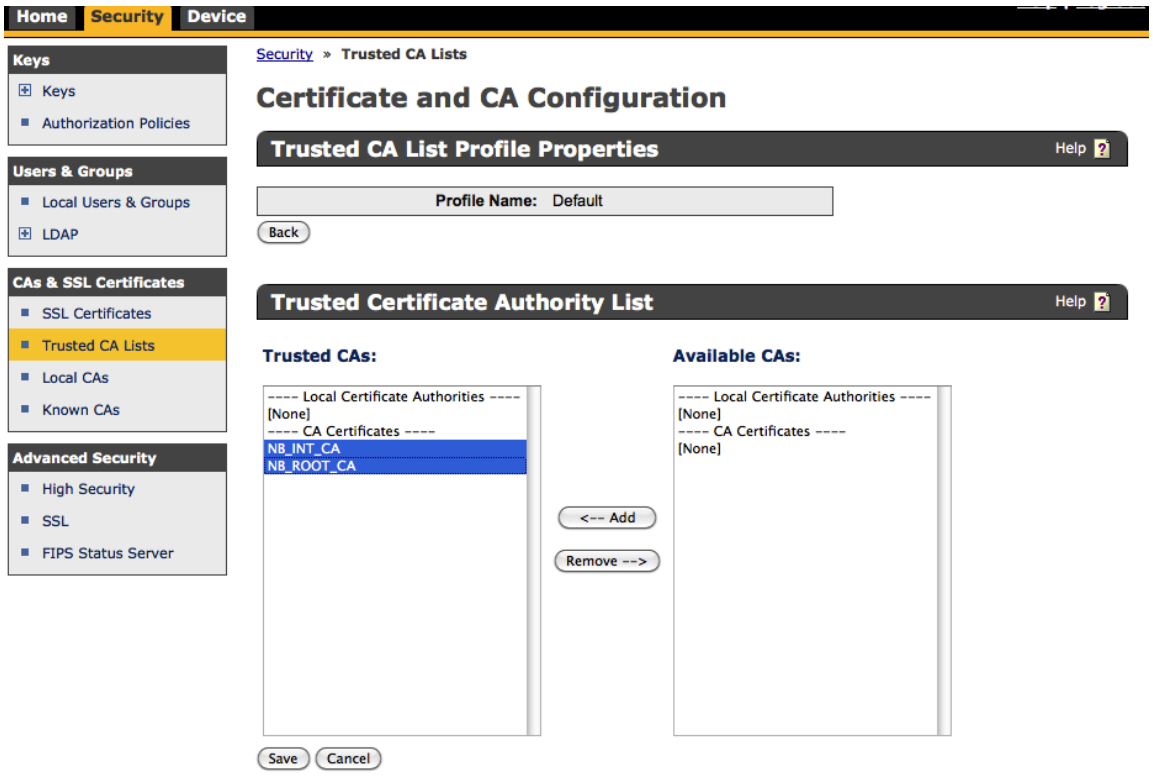


Figure 32) Save trusted CA list profile.

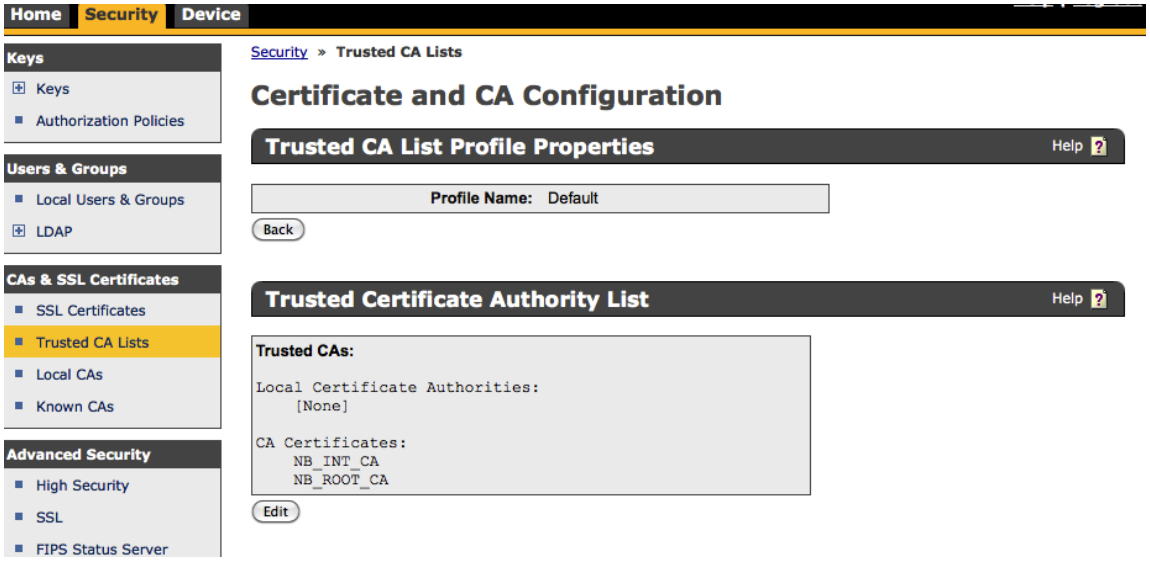


Figure 33) Download the Root CA public certificate

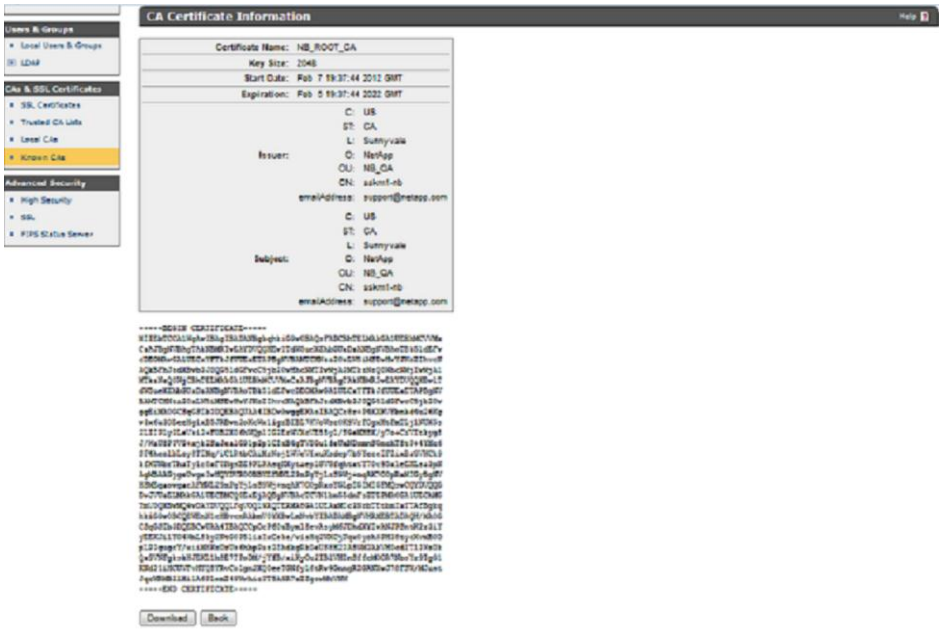
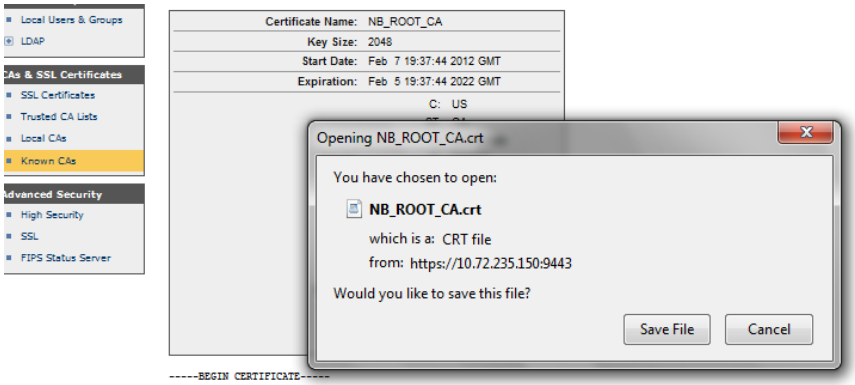


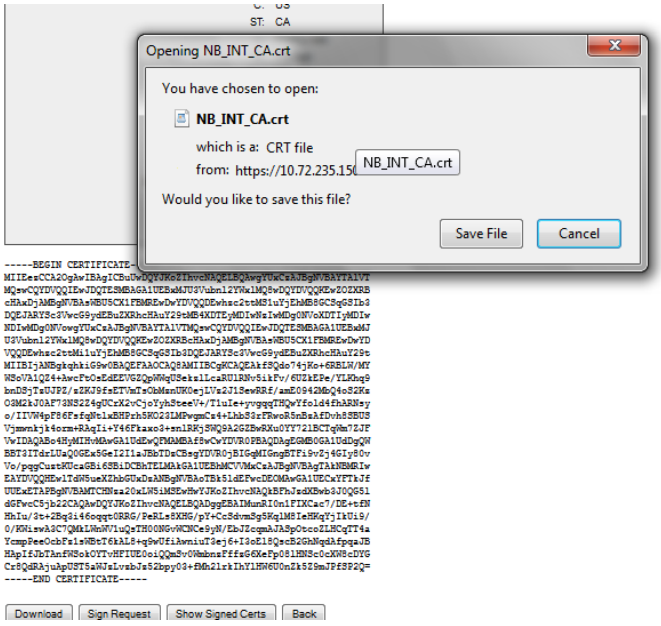
Figure 34) Save Root CA certificate



Alternatively you can also select the text from ----BEGIN CERTIFICATE…END CERTIFICATE---- and copy/paste it into a notepad/wordpad/text file and save it

Figure 35) Download Intermediate CA certificate

Figure 36) Save Intermediate CA certificate



Alternatively you can also select the text from ----BEGIN CERTIFICATE…END CERTIFICATE---- and copy/paste it into a notepad/wordpad/text file and save it

Create <IP_Address_KMIP_Server>_CA.pem File

This file is actually the public certificate of the CA server used to sign both the NSE and KMIP server .CSR files. If you're using a single CA, the file simply needs to be renamed <IP_ADDRESS_OF_KMIP_SERVER>_CA.pem. If there are two CA servers (root CA and intermediate CA), the files need to be concatenated together and then renamed.

In the example above we will concatenate the Root CA and Int CA pem files

```
root@core-vm30:~# cat CA_Root.pem CA_Int.pem >
<IP_ADDRESS_KMIP_SERVER>_CA.pem
```

At this point we have successfully imported the external CA servers (root CA and intermediate CA) into KeySecure, added them to the trusted CA list and have generated the <IP_Address_KMIP_Server>_CA.pem file

3.2.2.2. Signing CSR files by External CA server

In this section we will provide a general guideline that can be used by the External CA to sign the KeySecure CSR and NSE CSR

The External server in use can be customer's already existing external CA or it could also be a third party CA. Whatever the external CA server been used here, it is important that the External CA signs both the CSR files – KeySecure CSR and NSE CSR. Once both the CSR files are signed by the external CA, the signed CSR files need to be imported into KeySecure.

The signed KeySecure CSR needs to be in Server base 64 encoced X.509 format

The signed NSE CSR needs to be in Client base 64 encoded X.509 format

3.3 Create PEM Files for NSE

At this point, the Local CA / External CA has successfully signed the KeySecure CSR and NSE CSR. We now have the signed KeySecure CSR and signed NSE CSR and CA public certificate (<IP_Address_of_KMIP_Server>_CA.pem)

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:~#
```

At this point you've now created all the necessary SSL certificates needed for NSE and the KeySecure KMIP server.

3.4 Copy Keys and Certificates to NSE

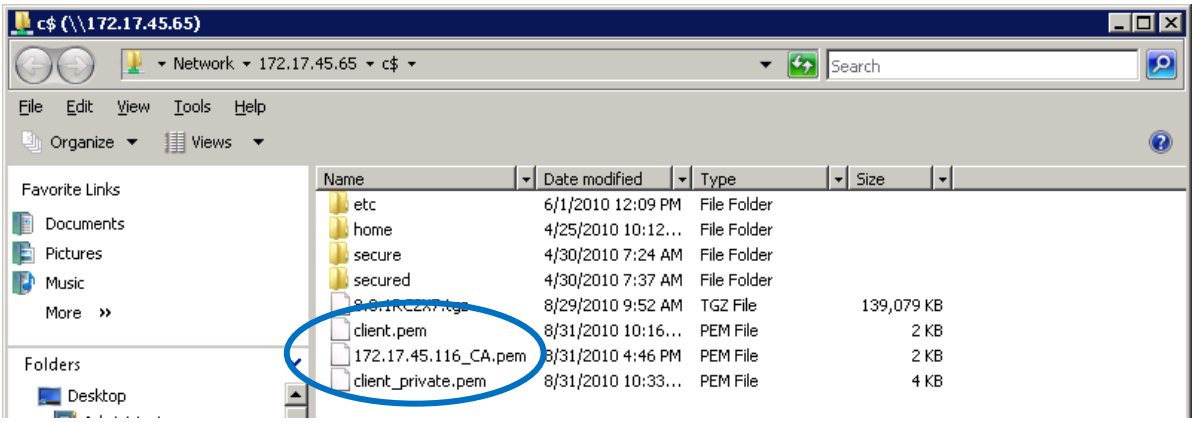
We now have a signed KeySecure CSR, signed NSE CSR (client.pem), NSE private key (client_private.pem) and CA public certificate (exported public certificate from the CA - <IP_Address_of_KMIP_Server>_CA.pem)

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/. where 172.17.45.65 corresponds to the ip address of the FAS platform that contains NSE drives. In this example we are going to use 172.17.45.116 as the ip address of the Local/External CA

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 KMIP 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 37) Moving PEM files on NSE volume.



3.5 Create KMIP Cryptographic Key Server Configuration

At this point we have successfully moved client.pem, client_private.pem and <IP_Address_of_KMIP_Server>_CA.pem files on the NSE volume (/vol/vol0)

This section will take you through the KMIP Cryptographic Key Sever configuration

This step configures KeySecure for communication using KMIP. It specifies the KMIP port to be used as well as the SSL certificate for the KMIP server.

Figure 38) Cryptographic key server configuration.

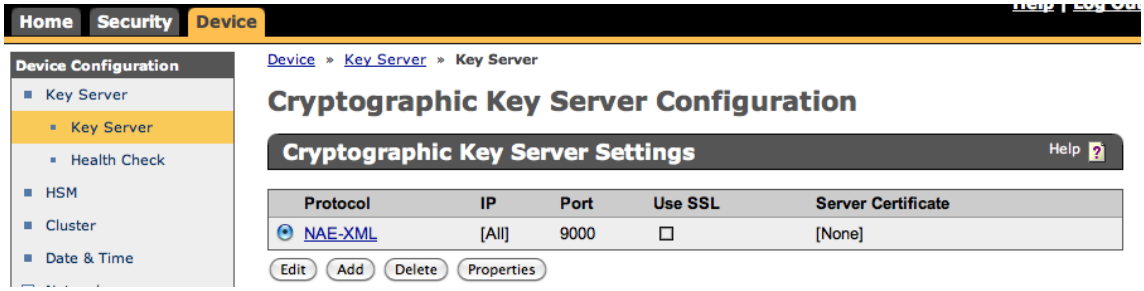


Figure 39) Add KMIP and specify port and server certificate.

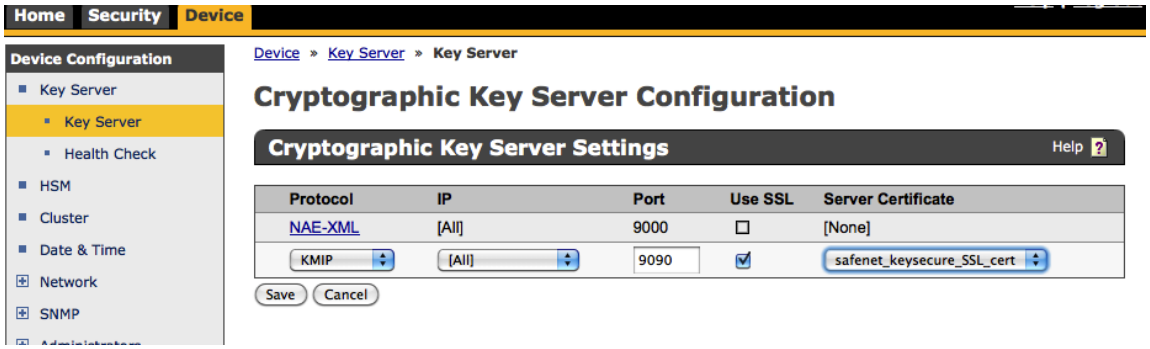
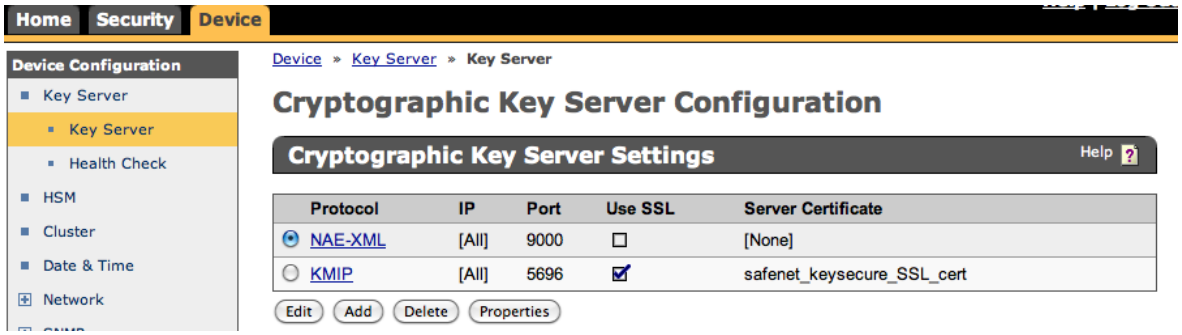


Figure 40) Add KMIP and specify server certificate.



4 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 >:<KMIP_Port> -
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 172.18.119.78:9090 -
verify 10 -showcerts -cert client.pem -key client_private.pem -CAfile
172.18.119.78_CA.pem

verify depth is 10

CONNECTED(00000003)

depth=2 /C=US/ST=CA/L=Sunnyvale/O=NetApp/OU=NB_QA/CN=sskm1-
nb/emailAddress=support@netapp.com

verify return:1

depth=1 /C=US/ST=CA/L=Sunnyvale/O=NetApp/OU=NB_QA/CN=sskm2-
nb/emailAddress=support@netapp.com

verify return:1

depth=0 /C=US/ST=CA/L=Sunnyvale/O=NetApp/OU=NB_QA/CN=safenet-ks460-
1/emailAddress=support@netapp.com

verify return:1

---

Certificate chain

 0 s:/C=US/ST=CA/L=Sunnyvale/O=NetApp/OU=NB_QA/CN=safenet-ks460-
1/emailAddress=support@netapp.com

   i:/C=US/ST=CA/L=Sunnyvale/O=NetApp/OU=NB_QA/CN=sskm2-
nb/emailAddress=support@netapp.com

-----BEGIN CERTIFICATE-----

MIIDrzCCApegAwIBAgICDzMwDQYJKoZIhvcNAQELBQAwgYUxCzAJBgNVBAYTAlVT

MQswCQYDVQQIEwJDQTESMBAGA1UEBxMJU3Vubnl2YWxlMQ8wDQYDVQQKEwZOZXRB

cHAxDjAMBgNVBAsWBU5CX1FBMREwDwYDVQQDEwhzc2ttMi1uYjEhMB8GCSqGSIb3

DQEJARYSc3VwcG9ydEBuZXRhcHAuY29tMB4XDTEyMDIxMjE4MjQxMloXDTIyMDIw

MjE4MjQxMlowgYwxCzAJBgNVBAYTAlVTMQswCQYDVQQIEwJDQTESMBAGA1UEBxMJ

U3Vubnl2YWxlMQ8wDQYDVQQKEwZOZXRBcHAxDjAMBgNVBAsWBU5CX1FBMRgwFgYD

VQQDEw9zYWZlbmV0LWtzNDYwLTExITAfBgkqhkiG9w0BCQEWEnN1cHBvcnRAbmV0

YXBwLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALck6cCPzPYF

d1Fs3ZxQX8BhaLfrr5iFYVERmSjtjN+XEi7VCFXsoYxg06IvDMZsF0v//rTbXF1K

HkLnD6gJso5sWUYsYiEAeokx7Pf2SqyeMM3KQV5Ryad1p7z11LiQdazjM0vuzBmg

nBUxGN35E8L8Mgsb/MIe7TIXK9gi6tUgoJqbALX4SWw21F5AWVzfVncw2EqCHCdJ

XAZI45rUxTSHAwsrmw8tZ7OFHpAOpl2kmmgCqKTVF7jVFvz27ZmMJSKm6SWHv4kR

lOdgQP9LbS9DVYEMvA8SQiEX9oE44aDGsWiujGpmCKBOJW1iZT2E0ILWdlMtECG8

wYsnCoWZg80CAwEAAaMgMB4wCQYDVR0TBAIwADARBglghkgBhvhCAQEEBAMCBkAw

DQYJKoZIhvcNAQELBQADggEBAGItGgHTYlQF6mmy+b+ubBdwseZToqYYBllg9ByT

aw6nDI8DrAnRn+g8wihw7nSg3VlsNO6Y+3dt0XJZ0UvTNrmsafHc+eGzncJRDlXr

npdzsUdNOdGSG/6MXhKEkb5KN0gJWmulLg61vJuUOL9SDMOu/RqWoW+H5GQZ71OW

3NmZcZ10UG7BHjtL+sPBa9iyQC4WmeXDOWtB8GwppPYSBADHudxxz3WbqAoUV+02

EYsLS01Rm0LYE7Lf3b3gW8/uvREYo4r6kBIMc+n2N4RohbbDtp81djlL5pvPKF9D
```

```
zucEj7gKjOy+aBYgvYkJ/njPR0ot1n5HtPaXiMZuDVyamqc=

-----END CERTIFICATE-----

---

Server certificate

subject=/C=US/ST=CA/L=Sunnyvale/O=NetApp/OU=NB_QA/CN=safenet-ks460-
1/emailAddress=support@netapp.com

issuer=/C=US/ST=CA/L=Sunnyvale/O=NetApp/OU=NB_QA/CN=sskm2-
nb/emailAddress=support@netapp.com

---

No client certificate CA names sent

---

SSL handshake has read 1263 bytes and written 427 bytes

---

New, TLSv1/SSLv3, Cipher is AES128-SHA

Server public key is 2048 bit

Secure Renegotiation IS supported

Compression: NONE

Expansion: NONE

SSL-Session:

    Protocol  : TLSv1

    Cipher    : AES128-SHA

    Session-ID:
C0967515F2D223C798EE0EC01143F83C94F670284CD8409BD27D8CA0B54BC485

    Session-ID-ctx:

    Master-Key:
3A5F4E82FCD60526D5349CA0760358E4F4FC145BB6261960FE96A5917B11FDB8659ED8AE248E2
F7152B2F5FA14CD4BFB

    Key-Arg   : None

    TLS session ticket:

    0000 - 7c 9b 99 b2 c7 be 50 8e-8b 6e 78 69 95 fe 24 55   |.....P..nxi..$U

    0010 - 87 ab 3c f5 19 fb e4 33-c2 bc 4e a3 72 27 84 6c   ..<....3..N.r'.l

    0020 - d6 7b 6e 34 99 3e 58 87-b1 62 af cb 7d ab 0a af   .{n4.>X..b..}...

    0030 - 58 ee 15 01 9c 0c 59 7e-26 cc 55 48 ad ce 15 06   X.....Y~&.UH....

    0040 - 2b 33 e2 b8 58 d5 4b d8-ac ef 54 ef 5e 5d a0 6e   +3..X.K...T.^].n

    0050 - d6 6c d3 c9 05 0f 76 9e-36 f4 2e 6f b8 37 f7 7c   .l....v.6..o.7.|

    0060 - f9 72 2f 84 3a 26 14 5e-a4 57 69 21 b0 62 2b 30   .r/.:&.^.Wi!.b+0

    0070 - c0 eb 06 92 aa d1 62 15-f6 c3 42 1b 4a 8b 21 ca   ......b...B.J.!.

    0080 - d2 c3 f8 0b 31 00 e9 e9-db d2 db c3 58 f3 c3 cd   ....1.......X...

    0090 - 1f 3f 35 e8 5a 59 7c e2-e0 ec c6 b0 3f 6f ac 93   .?5.ZY|.....?o..
```

```
    Start Time: 1329157511
    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. This is an example of successful output using three valid PEM files.

5 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 can 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.

6 KeySecure HA Cluster Pair Considerations

Lets look at the KeySecure HA Cluster Pair Considerations with the help of an example.

Assume that you have a NSE filer and SafeNet KeySecure k460 at the primary site and that you have ran through all the NSE preinstallation steps as mentioned in this guide and ran through the NSE setup. You also have a KeySecure at the DR site and you want the NSE filer at the primary site have the ability to talk to the KeySecure at the DR site and also make sure that the KeySecures are always synchronizing the keys between them. Here are the steps you need to follow in order for the NSE filer at primary site be able to talk to the KeySecure at the DR site

1. Populate the KeySecure on the DR site with the primary site CA in the "Known CAs" field .
2. Add the primary site CA under "Trusted CA lists" on the KeySecure on the DR site
3. Copy the IP_ADDRESS_KMIP_SERVER_CA.pem file of the CA from the DR site to the primary site NSE filer under NSE filer:/vol/vol0
4. Create the SSL KeySecure CSR on the DR site and have it signed by the CA on the DR site.
5. Configure KMIP on the KeySecure at the DR site for the KeySecure to be able to communicate to the NSE filer on the primary site
6. Run "keymgr install cert / IP_ADDRESS_KMIP_SERVER_CA.pem " on the NSE filer. Here IP_ ADDRESS_KMIP_SERVER_CA corresponds to the ip address of the CA on the DR site
7. Run "key_manager –add key_server IP_ADDRESS_KMIP_SERVER_CA" on the NSE filer using the ip address of the CA on the DR site. This step will make sure to register the CA on the DR site with the NSE filer
8. Run "key_manager query" to make sure that both the primary and secondary site Keysecures are responding to the NSE filer at primary site

9. In order to make sure that the KeySecures at the primary site and DR site are always automatically synchronizing the keys between them, make them part of the same cluster

By following the above steps you have now configured the NSE filer at the primary site to also have the ability to talk to the KeySecure at the DR site. This is very useful in situations when the primary site KeySecure goes down. You have also made sure that when both the Keysecures are up and running, they are automatically synchronizing the keys between them since they have been made part of the same cluster

Also remember that at any given time, NSE filer can talk to maximum of 4 Key Managers at a time.

Note: If you are running clustered Data ONTAP 8.2, 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. If you are running clustered Data ONTAP 8.2 , then the commands in step 6, 7 and 8 above need to be executed at the nodeshell level

APPENDIX

Certificate Cleanup

Removal of Existing Certificates

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.

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
keymgr view cert client_private.pem
keymgr view cert <IP_Address_of_KMIP_Server>_CA.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 the section "Certificate Cleanup" earlier in this appendix.
4. Install new certificates according to section 5, "Import 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.

Enabling FIPS Compliance Mode on KeySecure

FIPS compliance mode is an optional setting for the KeySecure with numerous requirements to make sure of a higher level of security. One mode that particularly affects NSE systems is the ability to make sure that only authenticated SSL certificates are received by the key management server.

Enable FIPS Compliance

Figure 41) FIPS compliance default status.

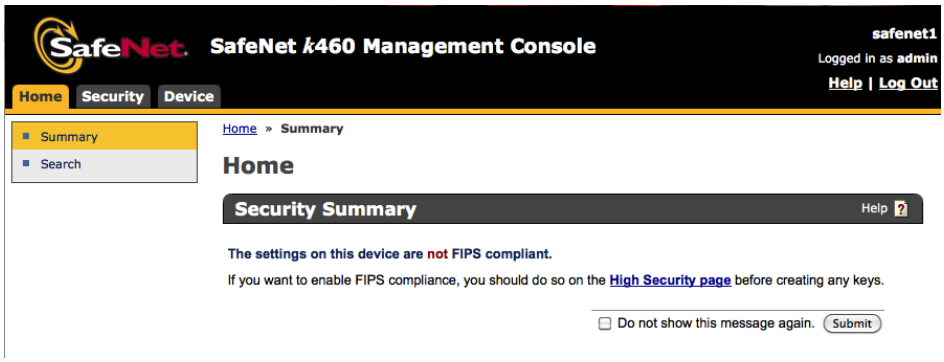
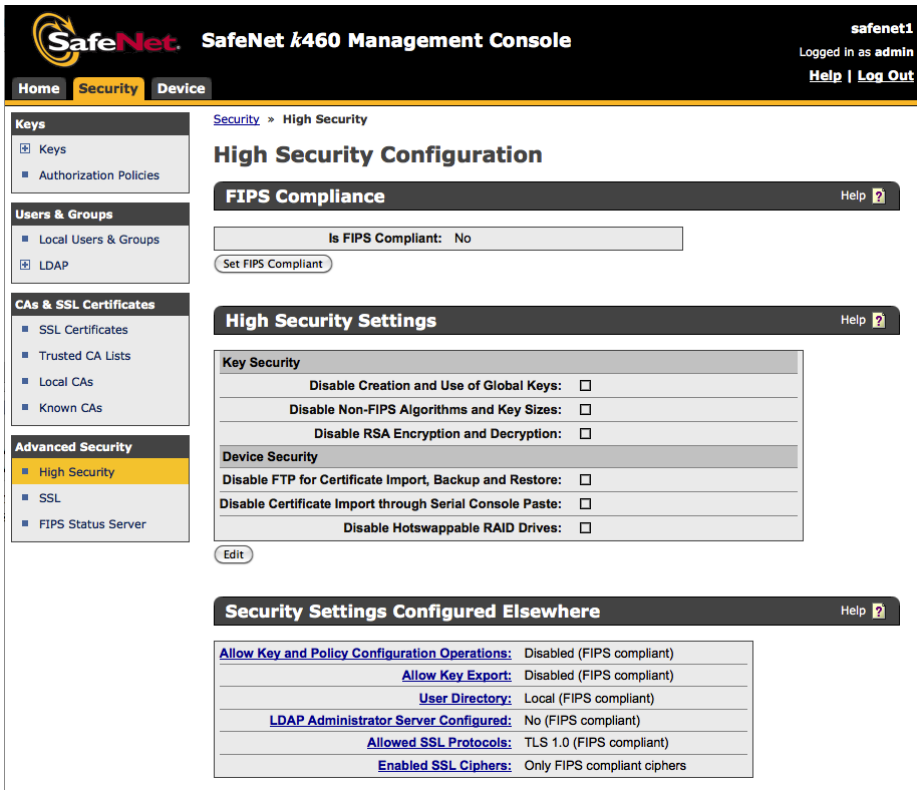


Figure 42) Enable FIPS compliance.



Creation of User Account

The user account here will match one specified field in the SSL certificate from the NSE client. In this example, CPOC is the organization unit (OU) in the SSL client certificate, and it will match the user account created here.

Figure 43) Creation of local user account for authentication.

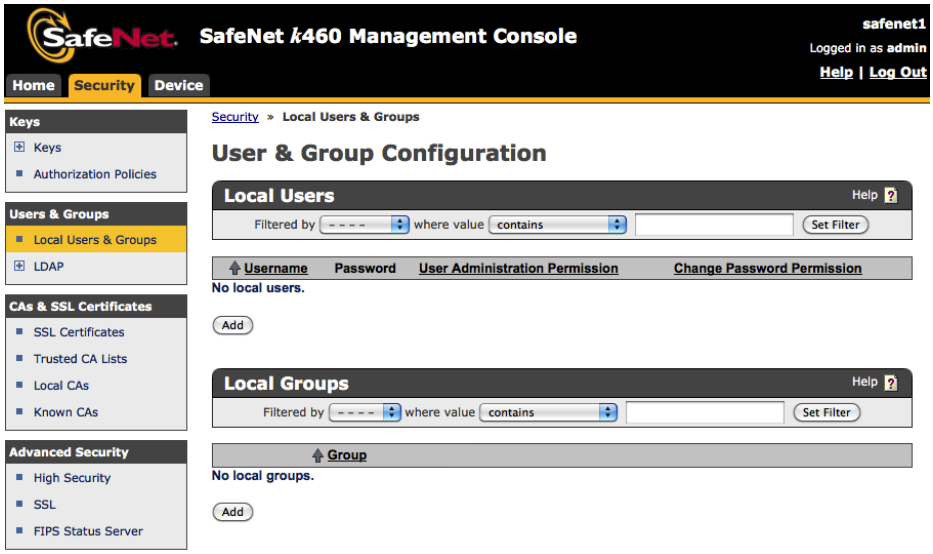


Figure 44) Specify user name and password.

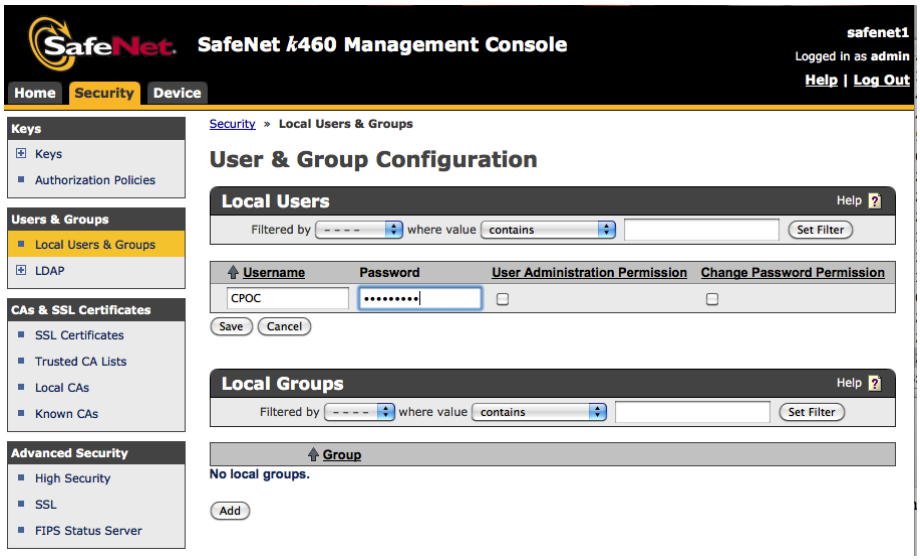
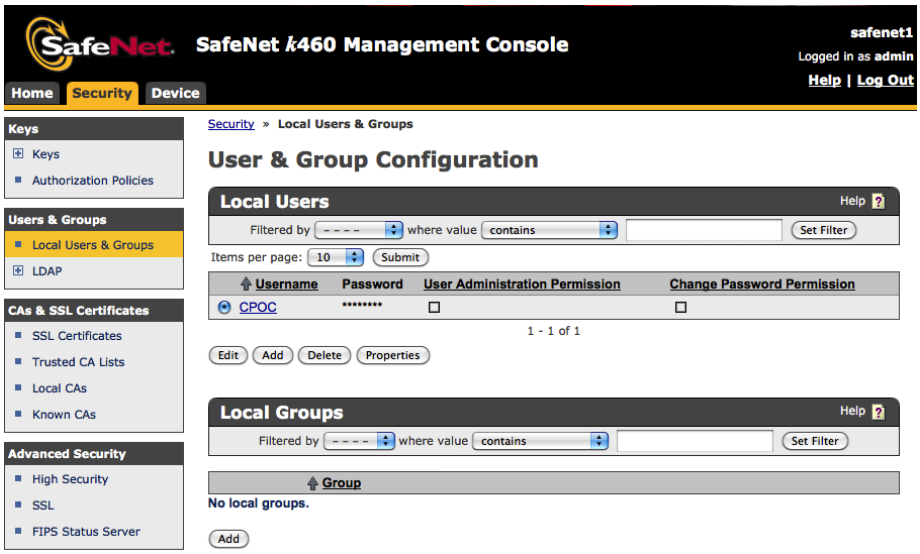


Figure 45) Save new user.



Configure Authentication Object in Cryptographic Key Server

Configuration of the specific object used to authenticate in the SSL certificate is performed on the Cryptographic Key Server Protocol. There are several choices, including common name, organizational unit, and email address. In this example, we are specifying the OU = CPOC, which needs to match the OU field in our NSE SSL client certificate.

Figure 46) Select KMIP.

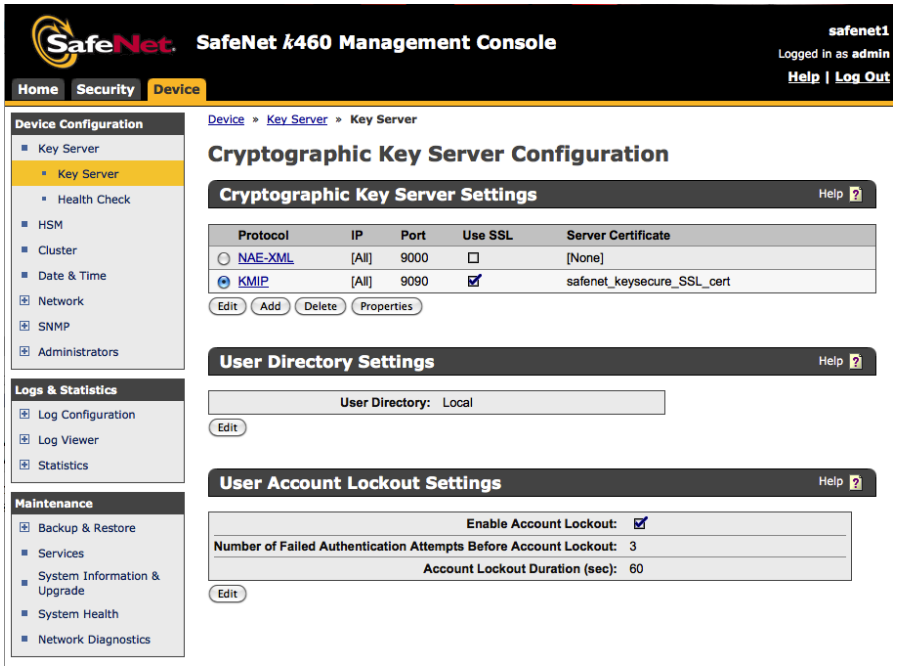


Figure 47) Edit authentication settings.

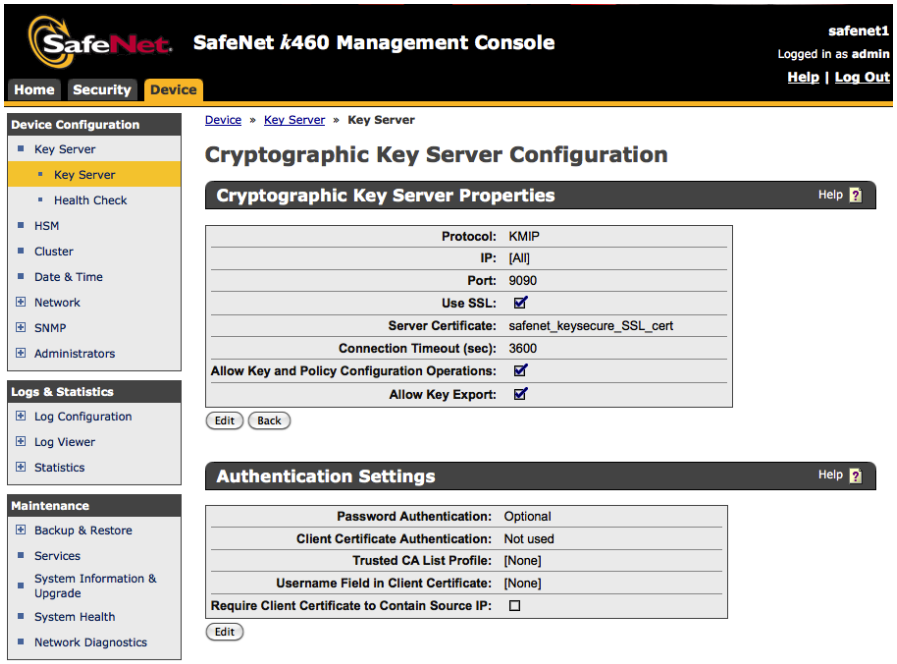
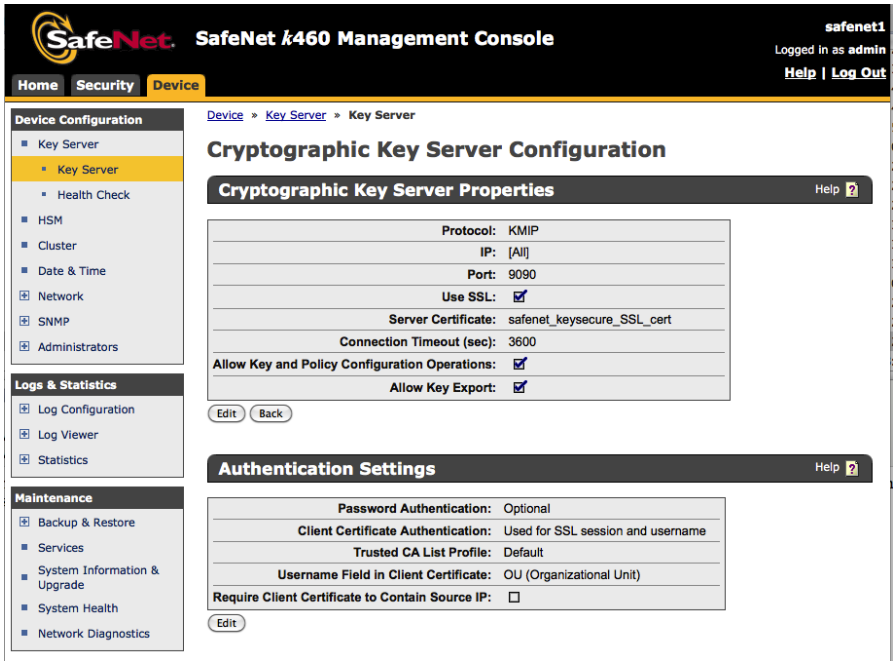


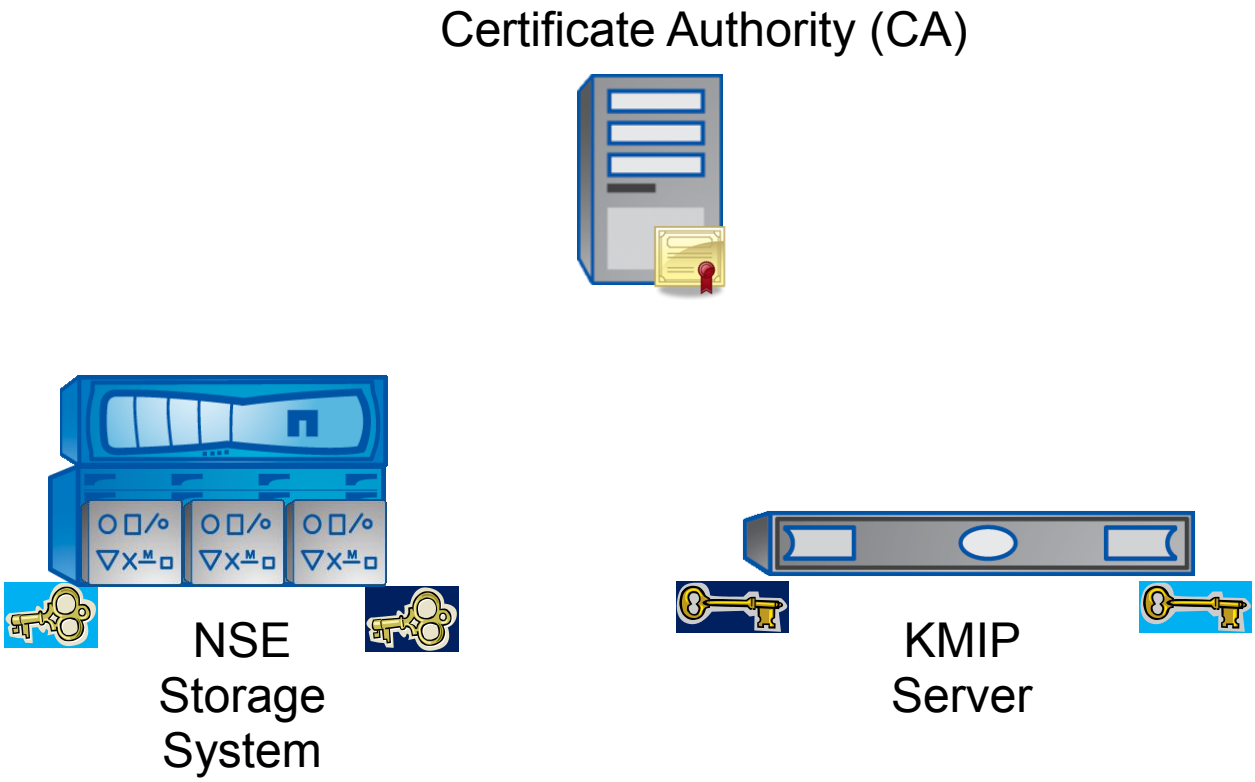
Figure 48) Specify user name field per SSL certificate parameters.

Figure 49) Save changes to authentication settings.



Certificates 101

Figure 50) Key generation and certificate authority.



A key exchange needs to occur between two entities, and a CA helps make sure of 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

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

www.netapp.com