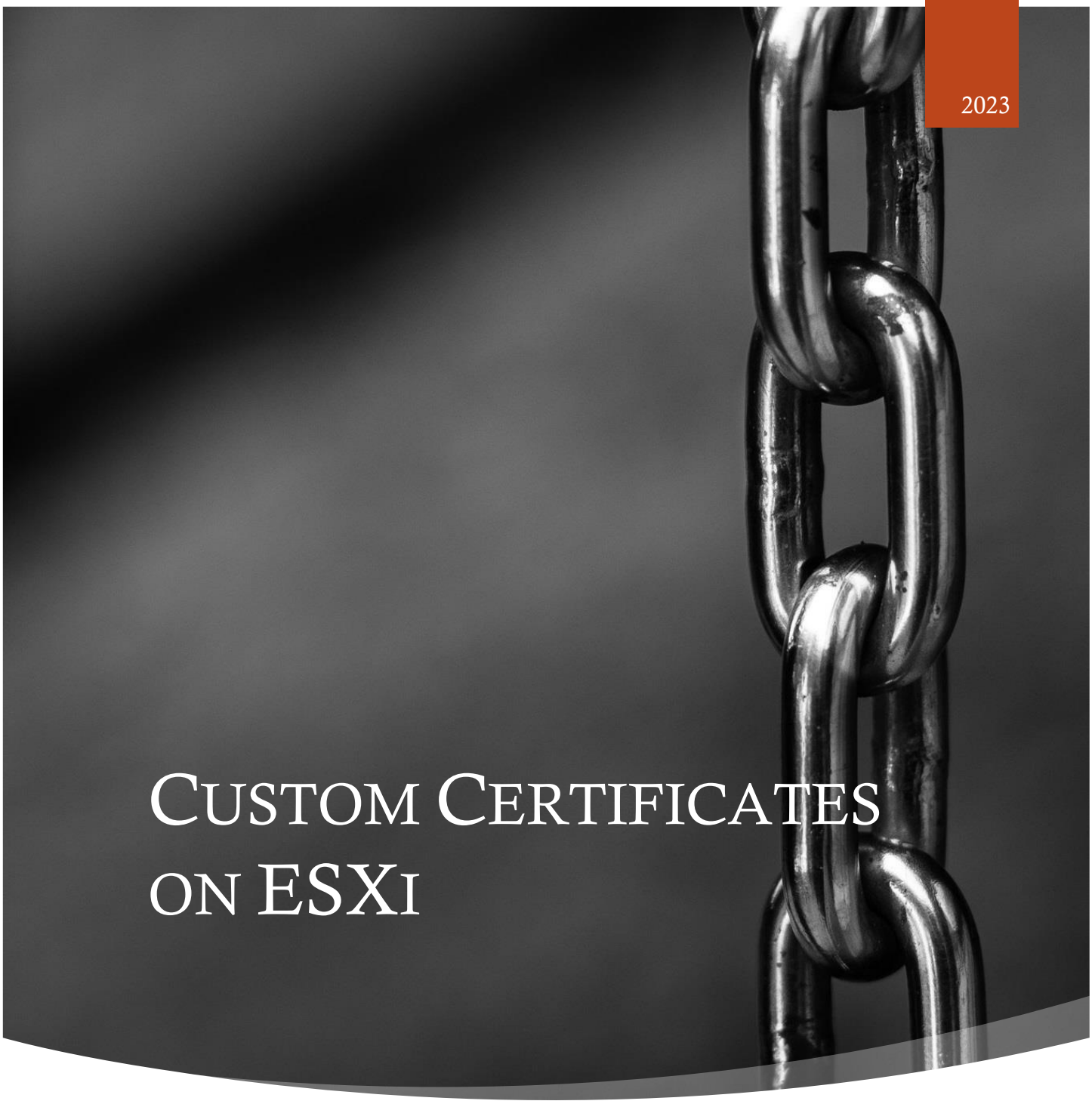
2023



Custom Certificates on ESXi

A DOD SECURITY REQUIREMENT

CLARK MERCER

Revision History

Date	Revision	Description
01/6/2020	0.90	Initial draft
02/11/2020	1.0	Added pkcs7 command and completed document
05/11/2020	1.1	Added pkcs8 portion since ESXi does not understand pkcs1
07/21/2020	1.2	Prefixed all occurrences of "openssl.exe" with ".\", since problems may be encountered when the path to openssl.exe is not given
9/10/2020	1.3	Added UTF8 encoding details to pkcs7 section
9/11/2020	1.4	Added "C:\Certs" and environment PATH sections to make commands simpler (what was ".\openssl.exe" is now "openssl")
12/14/2021	1.5	Added section for 'Certificate Renewals' and updated cover page
03/03/2022	1.6	Changed emails to '@army.mil', Updates to disconnect/reconnect from/to vCenter, and a few minor tweaks
05/11/2022	1.7	Updated Integral Files diagram
06/02/2022	1.8	Updated link in References for 'Step-by-step instructions'
08/15/2023	1.9	List alternate tool 'KeyStore Explorer'. Various minor updates.

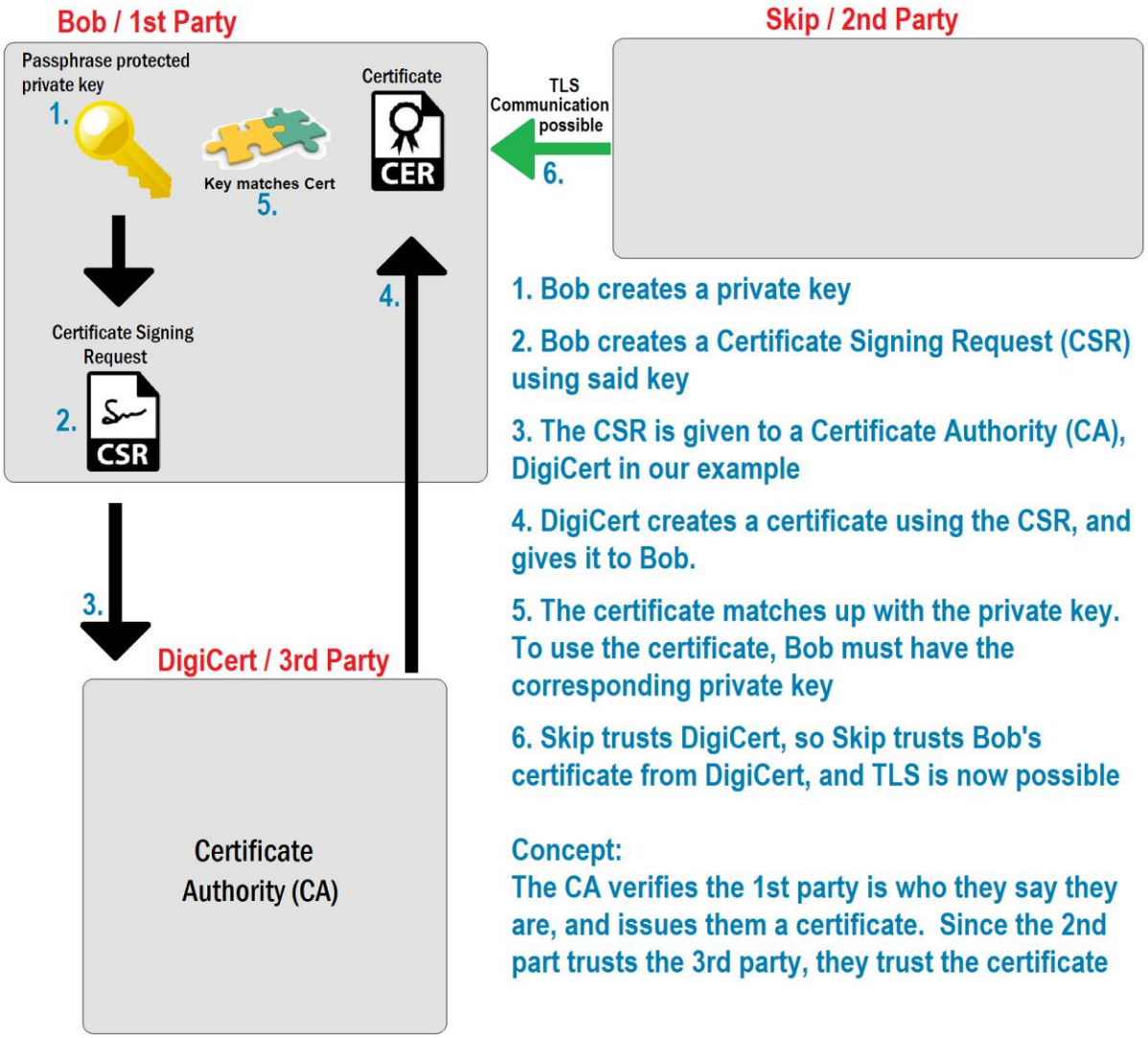
Contents

Overview

When you first install ESXi, it automatically generates a self-signed certificate. You may notice this certificate upon initial configuration when you point your browser to the host (e.g., https://10.120.243.223). Once an ESXi host is configured and connected to vCenter, this certificate may be replaced by one issued by vCenter since that is the default action in vSphere 6.0+. This document will provide guidance on replacing either the self-signed or the vCenter issued certificate with a custom 3rd party certificate, like one issued by a DoD Certificate Authority (CA).

Integral Files

The diagram below outlines the concept of digital certificates and may provide an understanding of the three different files that are integral to this concept – namely, key, CSR, and certificate.



A fundamental part of certificates is their expiration and renewal. Typically, certificates expire every one to three years. It must be considered an ongoing effort to maintain valid, non-expired certificates. A certificate should be regenerated shortly before the one it is replacing expires or as soon as possible thereafter.

For ESXi, the private key and Certificate Signing Request (CSR) are not generated within ESXi, but from a separate system. After you have submitted your CSR and received the certificate, the private key and certificate are transferred to the ESXi host where they will be used.

Procedure Summary

At a high level, the steps are:

1. From a secondary system, generate the private key and CSR
2. Use the CSR to obtain a certificate
3. Combine the certificate with the intermediate and root CA certificates
4. Place the ESXi host in maintenance mode and enable SSH
5. Disconnect the ESXi host from vCenter
6. Back up the pre-existing key and certificate
7. Transfer the key and full-chain certificate to the ESXi host
8. Remove the passphrase from the private key and ensure proper permissions and filenames
9. Restart the management agents on the ESXi host for the changes to take effect
10. Reconnect the host back to vCenter
11. Put the ESXi host back into service

VCenter Certificate Mode

vCenter can operate in one of three certificate modes. The "vmca" mode is the default, but when using DoD certs on ESXi, vCenter must operate in "custom" mode. This vCenter advanced setting is called "vpxd.certmgmt.mode", and changing it requires a restart of the vCenter server service. This should be done before you use custom certificates on any of your ESXi hosts. If needed, detailed information on this can be found in the VMware "vSphere Security" guide under the section "Certificate Management for ESXi Hosts".

Certificate Renewals

When a certificate is about to expire, a new certificate will need to be generated. In general, this entails the same procedure outlined in this document. Therefore, the same steps can be followed with the addition of these tips:

- It's a good idea to make a backup of the current cert and key before overwriting, so as a fallback plan, you could revert to the old ones.
- After the new files are in place, and permissions set, ensure you restart the management agents on the ESXi host, then disconnect and reconnect the host from/to vCenter. This process ensures vCenter uses and displays the newly updated certificate.

Step by Step

Generate the Private Key and CSR

This part cannot be done from ESXi, so you must do it from a secondary system. This can be done using various methods on Windows or Linux. Here are four possible methods:

Windows 10

- Download, install and use a "non-light" version of "Win64 OpenSSL" command line. This is the method we use in this guide.
- Download, install and use KeyStore Explorer. This is a graphical tool that can be used for creating and working with certificates and keys. It can be obtained here: https://keystore-explorer.org
- Certificates snap-in for MMC. This provides a nice way to create the CSR. However, the downside is that it's cumbersome to get your private key from Windows. To do so you must import the certificate from the CA, then export it and the key to a PFX file, then finally convert the PFX to PEM.

Linux (any modern distro)

- Use the command line "openssl" binary. The openssl package is available on any Linux distro. The commands are the same as shown in this document except for the PowerShell specific commands **`Select-String`** and **`Out-File`**. The Linux equivalents of **`grep`** and **`>`** can be used instead. The nice thing about this method is the "man" pages are available to help you with the commands. When using the Windows version, you must refer to the online "man" pages in the references section of this document.

On Windows we will use the command line tool "openssl.exe". When using these commands, you must reference the path to that executable. Alternatively, you can modify the Windows PATH environment variable, which will simplify your commands. The later method will be used in this document but is optional. Just understand that if you do not modify your PATH environment variable, you must always provide the path to "openssl.exe" instead of just typing "openssl".

1. Install OpenSSL for Windows. I recommend using the latest 64bit MSI full version (not the light version) available here: https://slproweb.com/products/Win32OpenSSL.html
   - Install to "C:\Program Files\OpenSSL-Win64", and when asked, store the OpenSSL DLLs in the "bin" directory.
   - NOTE: Like any other program, bugs, improvements, and security fixes are made from time to time. This program should be kept up to date or uninstalled when no longer needed.
2. Add `C:\Program Files\OpenSSL-Win64\bin\` to your PATH environment variable. You must restart your computer for the PATH change to take effect. This step is optional, but if skipped, you'll always need to use the path to "openssl.exe". Modifying your PATH environment variable can be done using the Windows GUI or PowerShell and is outside the scope of this document. However, steps to do so can easily be found online. The below article is one good example showing Windows 10 GUI instructions. https://www.howtogeek.com/118594/how-to-edit-your-system-path-for-easy-command-line-access
3. Open an elevated PowerShell window and make a "Certs" directory.

```
PS C:\> mkdir 'C:\Certs'
```

4. Copy the openssl.cfg file into this directory, and then change into that directory.

```
PS C:\> cp 'C:\Program Files\OpenSSL-Win64\bin\openssl.cfg'
'C:\Certs\openssl-custom.cfg'
PS C:\> cd 'C:\Certs\'
```

5. Open "openssl-custom.cfg" in a text editor like "Notepad++" to make the following edits.
6. We want to use "v3_req", so uncomment the line shown below (remove the hash from the beginning of the line). You will then have a line like this:

```
req_extensions = v3_req # The extensions to add to a certificate request
```

7. Next add the "_default" values for the distinguished name. Values for UT-ARNG are shown below with the lines in red. Hard coding these defaults will make it easy to create CSRs for multiple servers.

```
req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
countryName_default         = US
countryName_min             = 2
countryName_max             = 2

stateOrProvinceName               = State or Province Name (full name)
stateOrProvinceName_default       = Utah

localityName                = Locality Name (eg, city)
localityName_default        = Draper


0.organizationName          = Organization Name (eg, company)
0.organizationName_default = U.S. Government

# we can do this but it is not needed normally :-)
#1.organizationName         = Second Organization Name (eg, company)
#1.organizationName_default       = World Wide Web Pty Ltd

organizationalUnitName            = Organizational Unit Name (eg, section)
organizationalUnitName_default    = USA OU=PKI OU=DoD

commonName                  = Common Name (e.g. server FQDN or YOUR name)
commonName_max              = 64

emailAddress                = Email Address
emailAddress_default        = ng.ut.utarng.list.j6-sas@army.mil
emailAddress_max            = 64
```

8. Now scroll down and under "[v3_req]", add the line shown in bold red below. The other lines should already appear, but are shown to provide context:

```
[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = @alt_names

[ v3_ca ]
```

9. Now you must add the "alt_names" section that you just referenced. In this new section (called a 'stanza') you will list the Subject Alternative Names (SANs). The first SAN should be the same as your Common Name (CN). After that, other SANs like the IP address can be listed.
   This part should be done at the very end of the file. **When creating CSRs for additional ESXi systems, the last few lines of the file are the only ones that will need to change.** We will now add the lines shown below to the end of the file. Note: the CN for this example is "NGUTSVSGK80223.NG.DS.ARMY.MIL", and that is also the first SAN listed. We then add a SAN for the short name and one for the IP address. Your hostname and IP address will have the same form but will obviously be different than what is shown. Please use the values specific to the ESXi host that you are creating the certificate for.

```
[ alt_names ]
DNS.1 = NGUTSVSGK80223.NG.DS.ARMY.MIL
DNS.2 = NGUTSVSGK80223
IP.1 = 10.120.243.223
```

10. With the cfg file updated and saved, we will now create a 2048 RSA private key that will be used to create the CSR. Back in your PowerShell, execute the following command (all on one line ending in "2048"):

```
PS C:\Certs> openssl genrsa -out .\NGUTSVSGK80223.temp.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
.....................................................+++++
........+++++
e is 65537 (0x010001)
```

11. We need to convert the temp key from pkcs1 to pkcs8 format. This is so that ESXi will be able to read the key. The command will ask for a passphrase to encrypt with. Please use something compliant with password policy and remember the passphrase.

```
PS C:\Certs> openssl pkcs8 -topk8 -in .\NGUTSVSGK80223.temp.key -out
.\NGUTSVSGK80223.key
Enter Encryption Password:
Verifying - Enter Encryption Password:
```

12. Remove the temp pkcs1 key (we will only need the pkcs8 key)

```
PS C:\Certs> rm .\NGUTSVSGK80223.temp.key
```

13. Now check the private key using the following command. When prompted, enter your passphrase.

```
PS C:\Certs> openssl rsa -noout -check -in .\NGUTSVSGK80223.key
Enter pass phrase for .\NGUTSVSGK80223.key:
RSA key ok
```

14. If the check command completes without error and says "RSA key ok" you can assume it was created correctly and that you have the correct passphrase.

15. You can now generate the CSR by referencing the private key and the custom cfg file that was modified before that. The openssl command below is shown on two lines for clarity, but it is all on one line ending with "NGUTSVSGK80223.csr". You will immediately be prompted for the key passphrase. Then it will ask questions for the distinguished name (DN) such as Country and State. Because of the default values in the cfg file, these require no typing (shown in orange), just tap [ENTER]. The common name will need to be carefully typed in (shown in red) and should be the fully qualified domain name. The final two questions for challenge password and optional company name can be left blank.

```
PS C:\Certs> openssl req -new -sha256 -key .\NGUTSVSGK80223.key -config
.\openssl-custom.cfg -out .\NGUTSVSGK80223.csr
Enter pass phrase for NGUTSVSGK80223.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) [US]:
State or Province Name (full name) [Utah]:
Locality Name (eg, city) [Draper]:
Organization Name (eg, company) [U.S. Government]:
Organizational Unit Name (eg, section) [USA OU=PKI OU=DoD]:
Common Name (e.g. server FQDN or YOUR name) []:NGUTSVSGK80223.NG.DS.ARMY.MIL
Email Address [ng.ut.utarng.list.j6-sas@army.mil]:

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

16. Now that the CSR is created, view the CSR details to ensure that you did everything correctly. Use the below command to do so. When looking over the output, verify that both the "Subject Alternative Name:", and the "Subject:" fields look correct.

```
PS C:\Certs> openssl req -text -in NGUTSVSGK80223.csr | Select-String
"subject" -Context 1

          Version: 1 (0x0)
>         Subject: C = US, ST = Utah, L = Draper, O = U.S. Government, OU =
USA OU=PKI OU=DoD, CN = NGUTSVSGK80223.NG.DS.ARMY.MIL, emailAddress =
ng.ut.utarng.list.j6-sas@army.mil
>         Subject Public Key Info:
              Public Key Algorithm: rsaEncryption
                  Digital Signature, Non Repudiation, Key Encipherment
>             X509v3 Subject Alternative Name:
              DNS:NGUTSVSGK80223.NG.DS.ARMY.MIL, DNS:NGUTSVSGK80223,
IP Address:10.120.243.223
```

If you have problems verifying either the private key or the CSR, check that the filenames you are using are correct. Also, if the commands are not working, try typing them in by hand instead of using copy/paste and using [tab] completion wherever possible. If things are still not working, you may need to start over and generate a new key and CSR.

Use the CSR to obtain your certificate

To obtain the certificate, give the CSR and host information to a Trusted Agent (TA). If the CSR is sent in email, it needs to be encrypted. The TA will submit all the CSR information to a DoD Certificate Authority (CA). It can take the Registration Authority (RA) a day or sometimes longer before they issue the certificate.

Combine the Certificate with the Root CA Bundle

Once you receive your certificate, it's best practice to combine it with the intermediate and root CA certificates, so that it's a single file containing the full certificate chain. If you use the single cert instead of the cert plus full chain, Nessus may still report a vulnerability, but ESXi will still work. Therefore, this step is optional, but recommended. All certificate files can be saved in base64 format which are plain text files. These files can then be opened in any text editor such as Notepad++ on Windows and manually combined into a single new UTF8 encoded file called "NGUTSVSGK80223-Full.crt" for example. The order of the certificates in this file is critical. Your certificate must appear first (at the top of the file), the intermediate in the middle, and the Root-CA certificate must appear last (at the bottom of the file). The resulting text file will have the structure shown below. I have color coded and abridged it for example purposes. Your text will be longer since the example shown is abbreviated, but the order of the three certificates must be the same.

```
-----BEGIN CERTIFICATE-----
MIIHbDCCBlSgAwIBAgITTgAtHWHkeUo64SlRCAACAC0dYTANBgkqhkiG9w0BAQsF
[SERVER cert at the top - abridged here]
R+SV6/UyCZgNNr+Uv2qJcOfoNZplV1VV2S9rjOA7KUs=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIGhjCCBW6gAwIBAgITXgAAAAxwBzP92NNQwwAAAAAADDANBgkqhkiG9w0BAQsF
[SUBORDINATE1 CA cert in the middle - abridged here]
QggGOeWKnxLQpj92OJVICLioL7ZEA/5d1+hU1YgmcFk6+uLiA5SOSuPK
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDgjCCAmqgAwIBAgIQHuLxR40dqr5P5N6LfLsSbzANBgkqhkiG9w0BAQsFADBJ
[ROOT CA cert at the bottom - abridged here]
9I+sMIMz+62Pcf6zIjz0yRFOC568VD2FhpOgIRqpppf//rTu5jI=
-----END CERTIFICATE-----
```

There are several ways to accomplish this. One way to do it is to download the cert in pkcs7 format (which contains the full chain), save it as "cert.pkcs7", and use the command below to create "NGUTSVSGK80223-Full.crt".

```
PS C:\Certs> openssl pkcs7 -in .\cert.pkcs7 -print_certs | Out-File
.\NGUTSVSGK80223-Full.crt -Encoding utf8
```

Prepare the ESXi Host and Backup the Existing Key and Certificate

1. From this step on, all virtual machines (VMs) on the ESXi host must be migrated off or powered off, and the host should be put in maintenance mode. To place the host in maintenance mode, you can use vCenter or a web browser pointed directly at the host.
2. Once the host is in maintenance mode, start its SSH service and disable lockdown mode.
3. Disconnect the host from vCenter (right click host → Connection → Disconnect).
4. Login to the host as root with an SSH client like PuTTY.
5. The certificate and private key on ESXi are always stored as the following files:

Description	File
The certificate file	/etc/vmware/ssl/rui.crt
The private key file	/etc/vmware/ssl/rui.key

6. You will be replacing these files, but before doing so, make a backup copy of them to a different directory. You can do so with a command similar to this:

```
# cp -p /etc/vmware/ssl/rui.*  /vmfs/volumes/NGUTSVSGK8023-UC-BOOT/tmp/
```

Note: Copy them to a path that is different than "/etc/vmware/ssl" because files in that directory may be deleted without warning.

Replace the Certificate and Key

1. Copy the full chain certificate "NGUTSVSGK80223-Full.crt" (3 certs in 1 file you created), and the corresponding private key "NGUTSVSGK80223.key", to the ESXi host. You could use "WinSCP" to do this, or just use a browser pointed to the ESXi web interface GUI (e.g., https://10.120.243.223). If you use the GUI, browse the storage, and choose "Upload Files" to upload them to a temp directory.
2. Now from your SSH session, copy the two files into the ssl directory using the "cp" command like this:

```
# cp -p /vmfs/volumes/NGUTSVSGK8023-UC-BOOT/tmp/NGUT*  /etc/vmware/ssl/
```

3. Once the files are on the ESXi host, remove the passphrase from the private key and name the resulting file "rui.key". It will prompt for the passphrase, then output the key, overwriting the existing "rui.key".

```
# cd /etc/vmware/ssl
# openssl rsa -in NGUTSVSGK80223.key -out rui.key
Enter pass phrase for NGUTSVSGK80223.key:
writing RSA key
```

4. While still in the "ssl" directory, rename "NGUTSVSGK80223-Full.crt" to instead be "rui.crt", overwriting the existing certificate.

```
# mv NGUTSVSGK80223-Full.crt  rui.crt
```

5. Now ensure the permissions 400 on the key and 644 on the certificate. The key and cert will not work correctly if the permissions are not correct:

```
# chmod 400 rui.key
# chmod 644 rui.crt
```

6. You should now have replaced both the private key and certificate with a custom key and certificate. If desired, you can verify that the key has no passphrase (because ESXi requires that there be none), and that the key corresponds to the certificate. This is done by comparing the digests using the single command shown (the commands are all on one line, but shown on two lines for clarity):

```
# openssl rsa  -noout -modulus -in rui.key | openssl sha1 &&
  openssl x509 -noout -modulus -in rui.crt | openssl sha1
```

7. If everything is correct, you will not be prompted to enter a passphrase. The output should be two identical digests, which indicates a match between the key and certificate. Your output will have a format like the example, but obviously be different:

```
(stdin)= d06f77a82fbae8a66c54a096f465bdda
(stdin)= d06f77a82fbae8a66c54a096f465bdda
```

8. With the new files in place, and permissions set, restart the management agents on the ESXi host for the changes to take effect. Restarting the management agents can be done a few ways. It can be done by logging into the direct console interface, or it can be done from within your SSH session with the command:

```
# services.sh restart
```

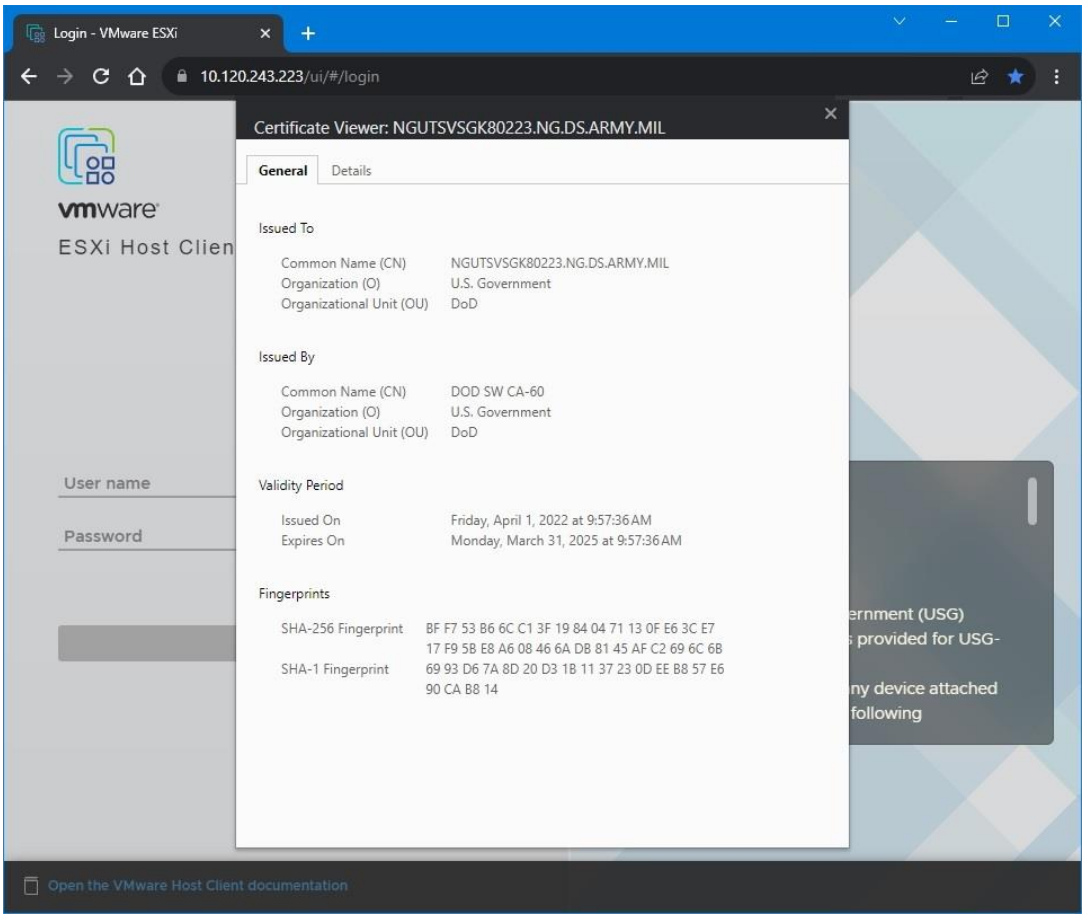
9. Reconnect the host to vCenter (right click host → Connection → Connect). If you find that disconnect/reconnect is not working or there is a problem, just remove the host from vCenter inventory, then add it back to vCenter.

Cleanup

Verify the new certificate on the ESXi host. If you cannot do this with a Nessus scan, you can at least view the trusted certificate by pointing your browser to the host address and examining the certificate. Using the Firefox browser, you could see mixed results because that browser uses a separate trust store. In the Chrome or Edge browser, you can click the padlock in the address bar, click 'Connection is secure', then click 'Certificate is valid' or the certificate icon. An example result using Chrome is shown on the next page. In the example, notice the padlock in the address bar.

Once the new certificate is verified, the ESXi host can be placed back into service.

1. Verify the new certificate.
2. Add the host back to vCenter if you have not already done so.
3. Exit all SSH connections to the host.
4. Stop the SSH service on the host
5. Re-enable lockdown mode on the host.
6. Exit maintenance mode.
7. Once the SSH service is stopped and the host is out of maintenance mode, the host can again run virtual machines.



Revocation of Certificates

Certificates need to be revoked if the corresponding private key is compromised or the server certificate is no longer needed, but there is no need to revoke expired certs. Once revoked, the certificates serial number will be added to the CRL or OCSP responder allowing TLS clients to check the validity of the certificate's status. Once a certificate has been revoked, an entirely new certificate will need to be issued to replace the revoked certificate. A TA will need to submit any needed revocations.

References

This document was written based on the authors experience combined with information available in the vSphere Security Guide.

vSphere Security Guide: "Certificate Management for ESXi Hosts" ..... Link Here
https://docs.vmware.com/en/VMware-vSphere/6.5/vsphere-esxi-vcenter-server-652-security-guide.pdf

Step-by-step instructions for creating a key & CSR on various system/application types ... Link Here
https://www.digicert.com/kb/csr-creation.htm

Man pages for openssl and subcommands.....Link Here
https://www.openssl.org/docs/man1.1.1/man1/openssl.html