Lab - Create Self Signed Certificate and Bind in IIS

Overview

In this lab, you will learn how to create a self-signed certificate and bind to a site in IIS A self-signed certificate is a certificate that is signed by itself rather than a trusted third party such as Microsoft or VERITAS.

By using self-signed certificates, no PKI (Public Key Infrastructure) needs to be deployed before/after deployment of server-side applications. However, using self-signed certificates has both advantages and disadvantages.

Advantages

1. No PKI (Public Key Infrastructure) is needed.

2. Automatic deployment (Usually Self-signed certificates created automatically during the installation process of the server-side applications).

Disadvantages

1. Other applications/operating systems will not trust the certificates. This may lead to authentications errors etc.

Note: To overcome this limitation, some IT staff add the self-signed certificates to the Trusted Roots Certificate Authorities. However, using this workaround may to additional time that needed for management and troubleshooting. **(Covered in this lab)**

2. Self-signed certificates lifetime is usually one year. Before the year is has ended, the certificate may need to renew/replace.

3. Self-signed certificates may use low hash and cipher technologies. Due to this, the security level that implemented by self-signed certificates may not satisfy the current Security Policy.

4. No support for advanced PKI (Public Key Infrastructure) functions (e.g., Online checking of the revocation list, etc.).

5. Most of the advanced feathers of server-side applications require a PKI (Public Key Infrastructure). Self-signed certificates can't be used.

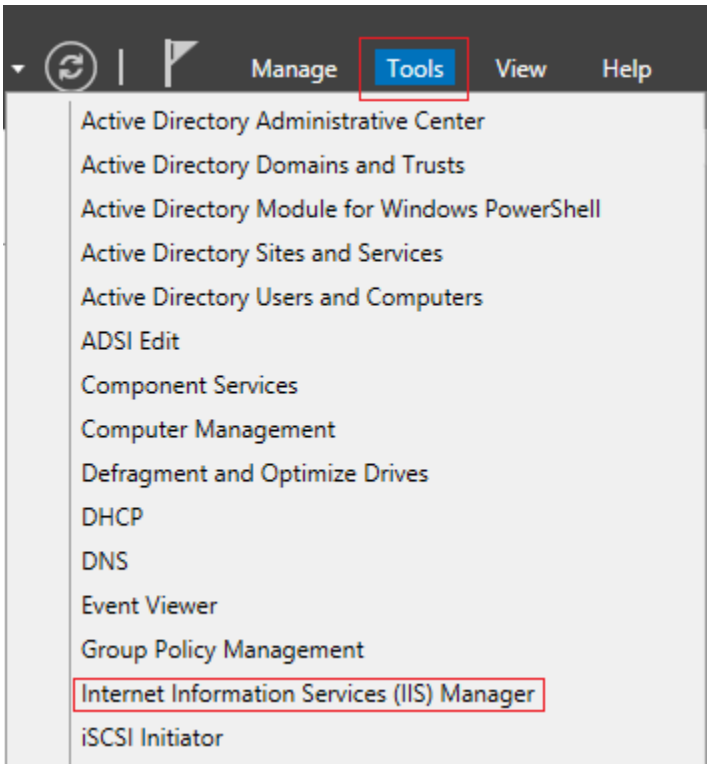
SYBEROFFENSE

Lab Requirements

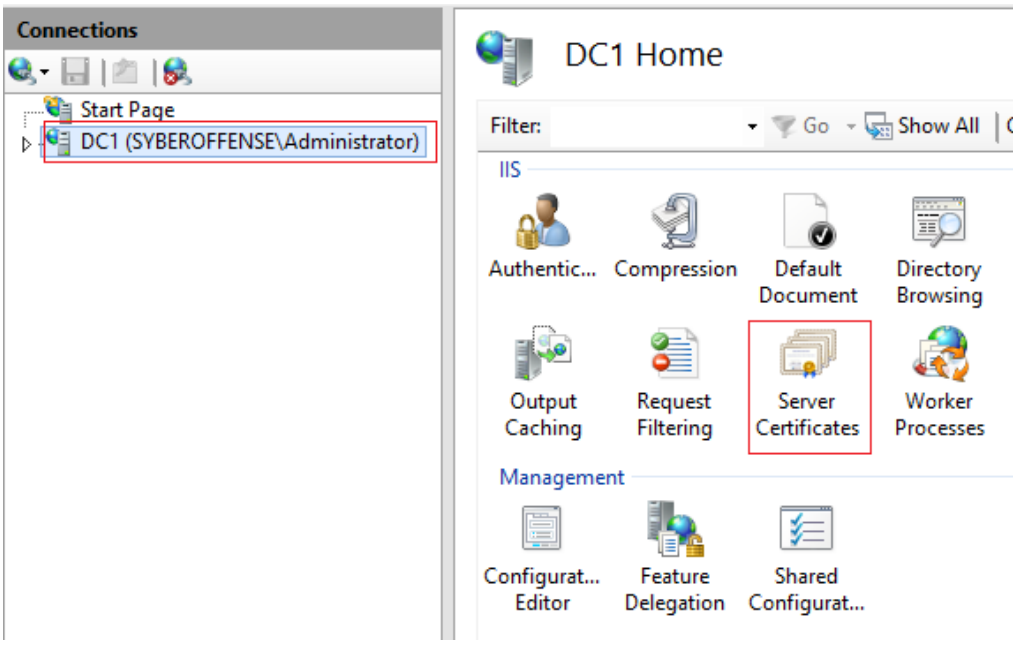
- One virtual install of Server 2012 or 2016 Full GUI running as a domain controller
- Web Services (IIS) installed

Begin the lab

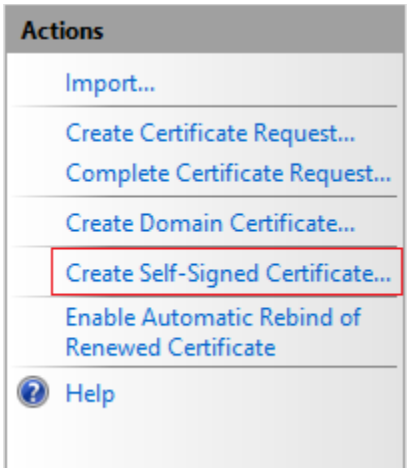
From Server Manager, go to the top right corner and click on tools. From the selection of snap-ins, Internet Information Services (IIS) Manager.



Inside the left window pane of your IIS management console, click on the name of your server. From the right window pane, click on Server Certificates.



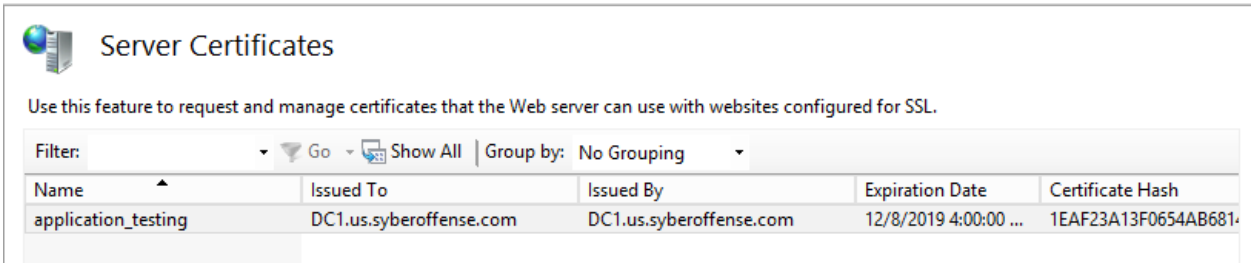
In the far-right window pane, click on Create a self-signed Certificate.



In the specify a Specify a Friendly Name window, type a name the signifies what the certificate will be used for. In this example, we are using this certificate for the testing of an application being developed in-house. This makes sense as we would not want to purchase a certificate from a trusted source just for testing purposes.

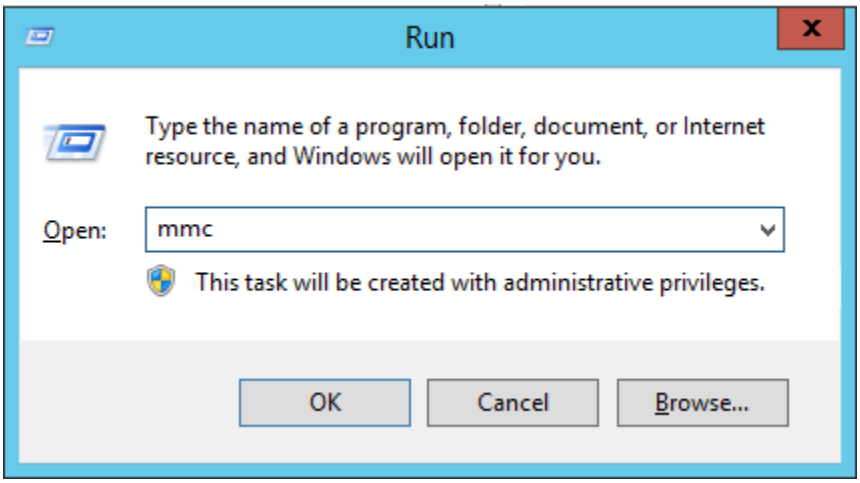
Leave the certificate store location as personal. If the certificate were to be used for hosting a website, we would change the location from personal to web hosting.

The certificate appears in the center window pane.

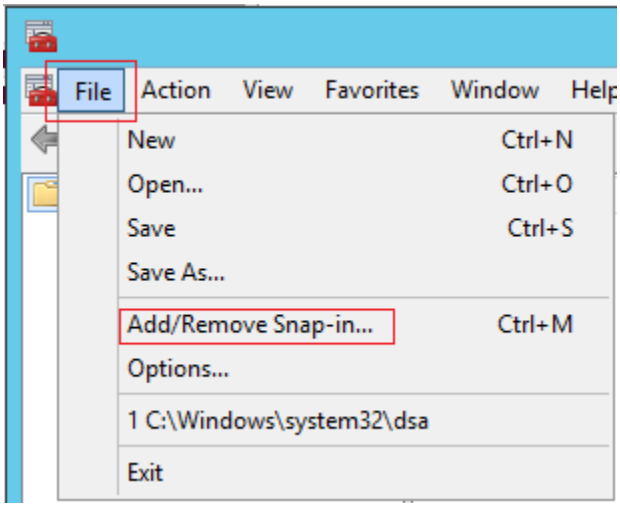


We can now go to the certificate store to view the certificate.

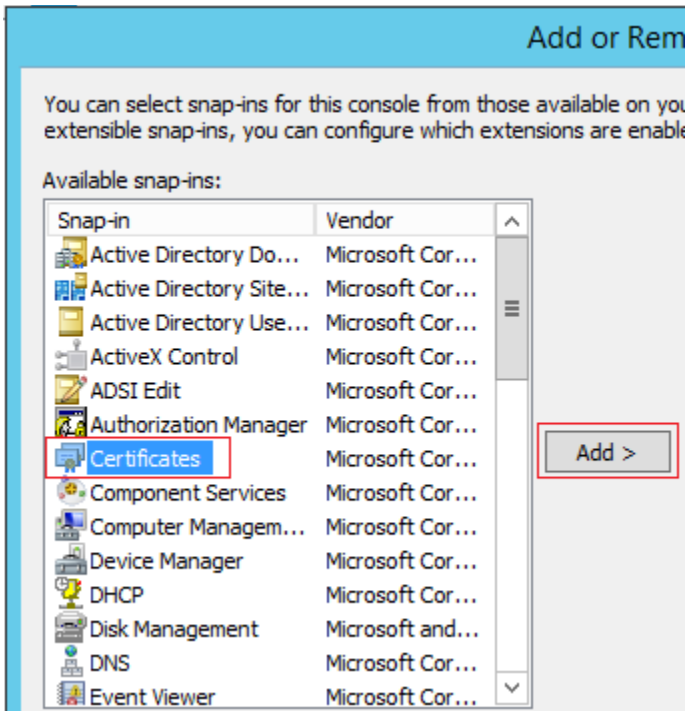
From your keyboard, press the Window+R key to open a run line. In the run line, type MMC, short for Microsoft Management Console. Click OK.



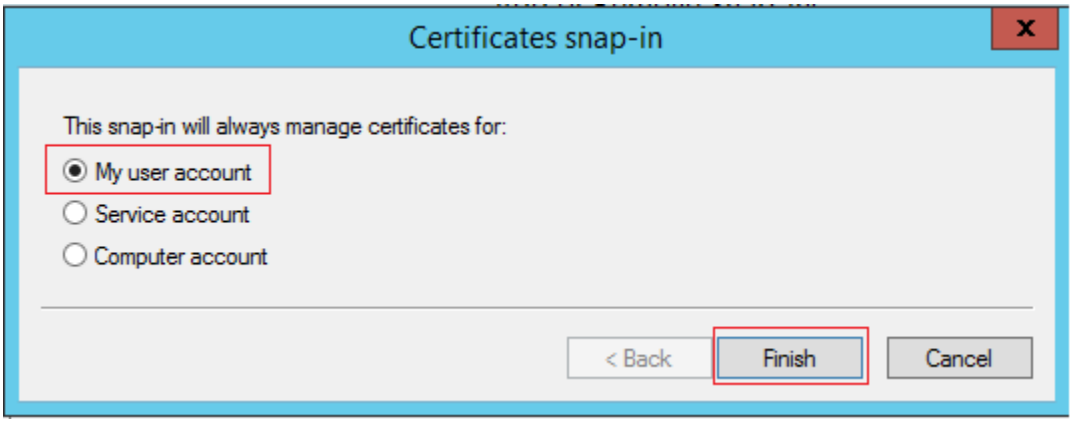
Click on File>Add/Remove Snap-in



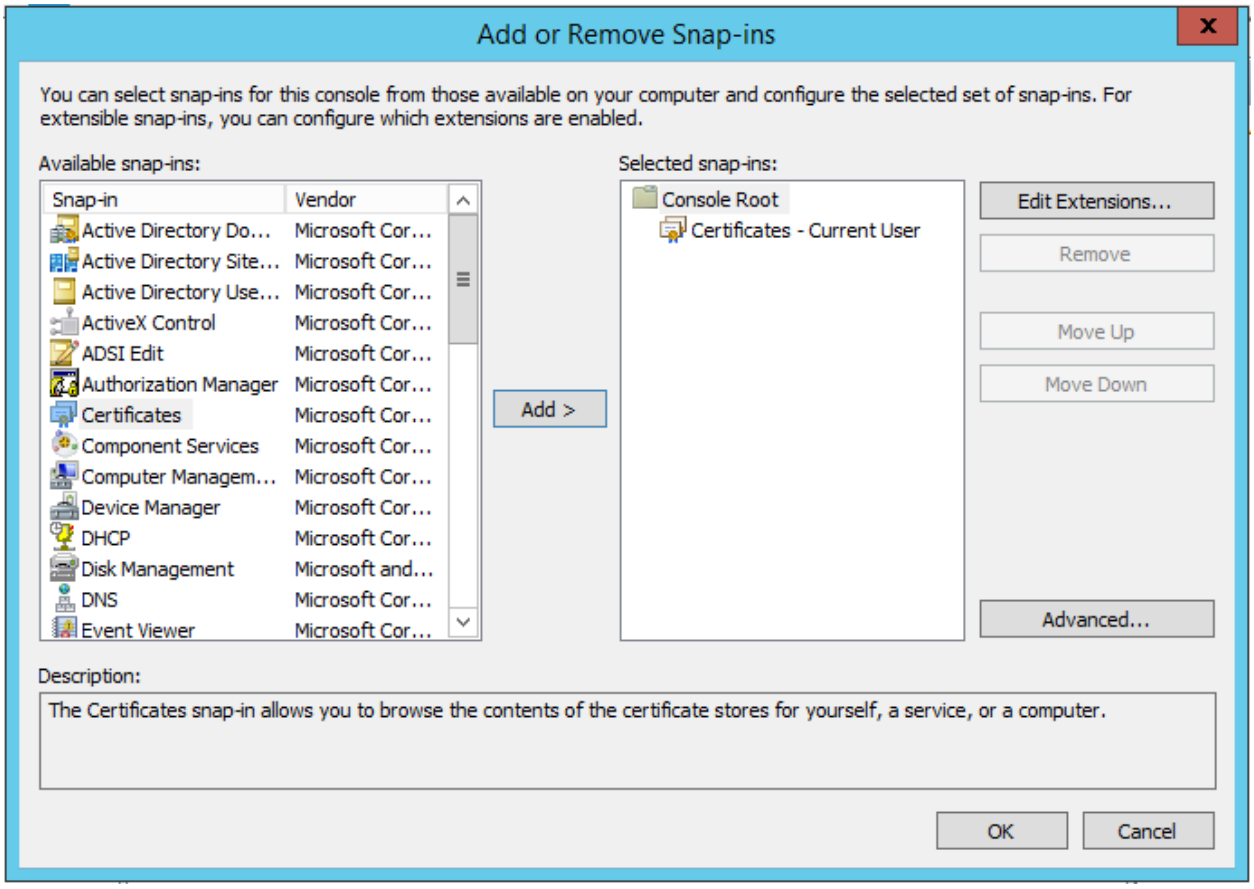
From the list of available snap-ins, click on Certificates and then the Add> button.



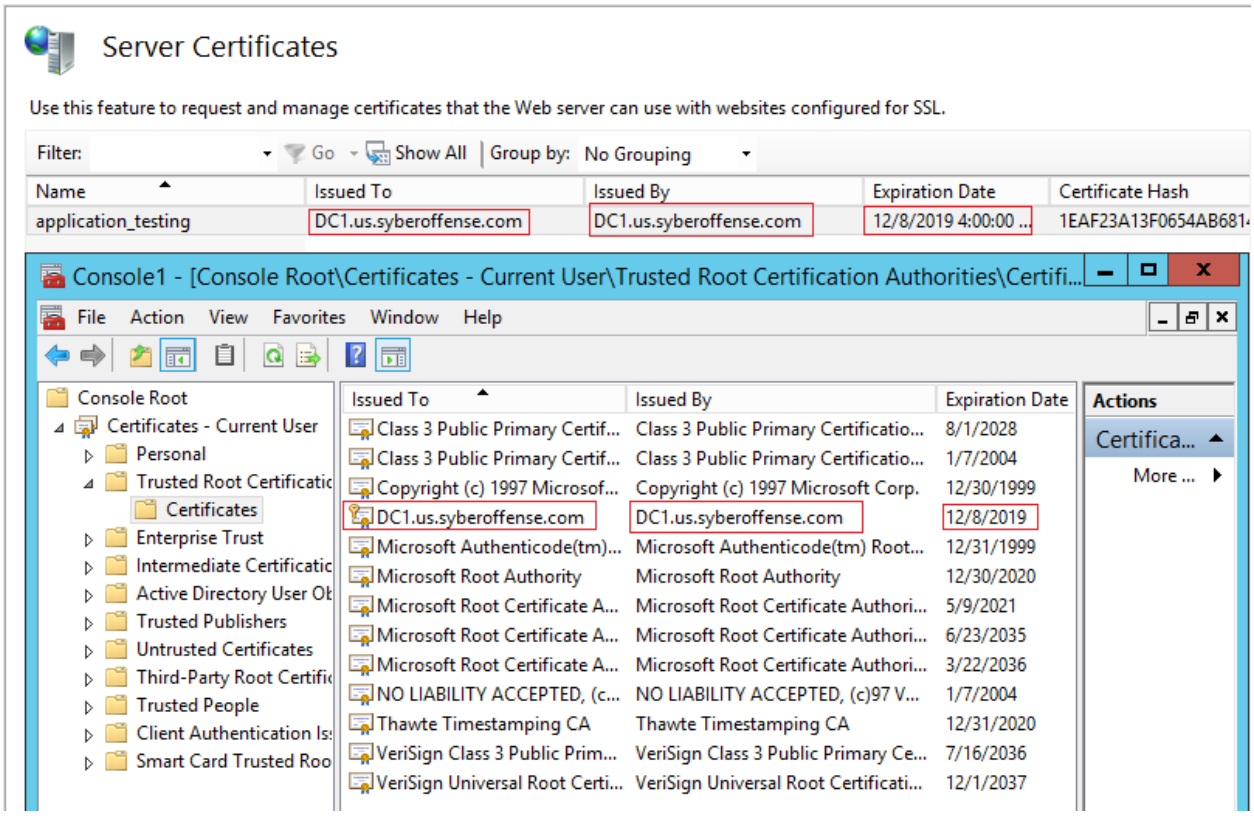
On this next screen, accept the default for My user account and click finish



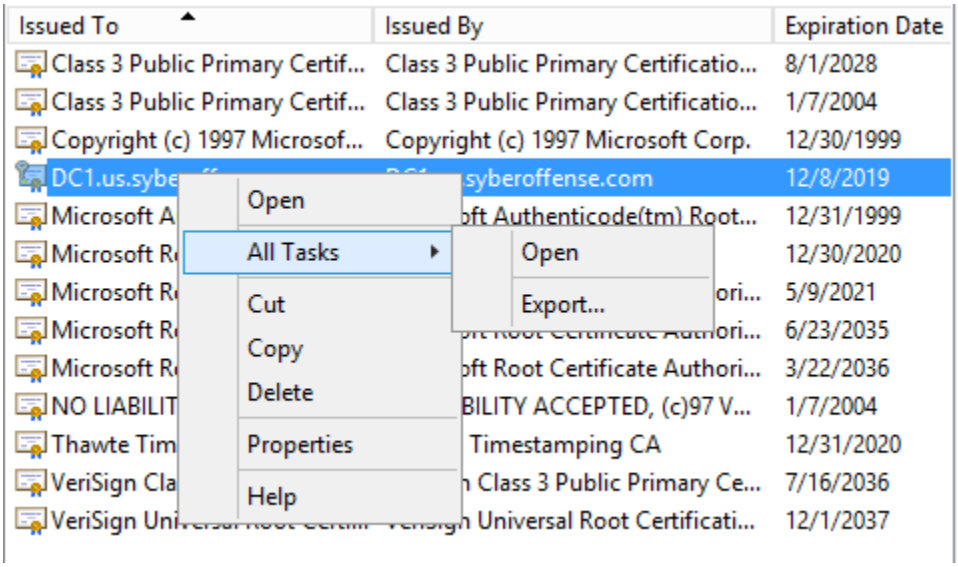
The snap-in is now ready to be added to the MMC console. Click OK.



Expand the Certificate-Current User and the Trusted Root container. In the center window pane, you can see the certificate we just created.



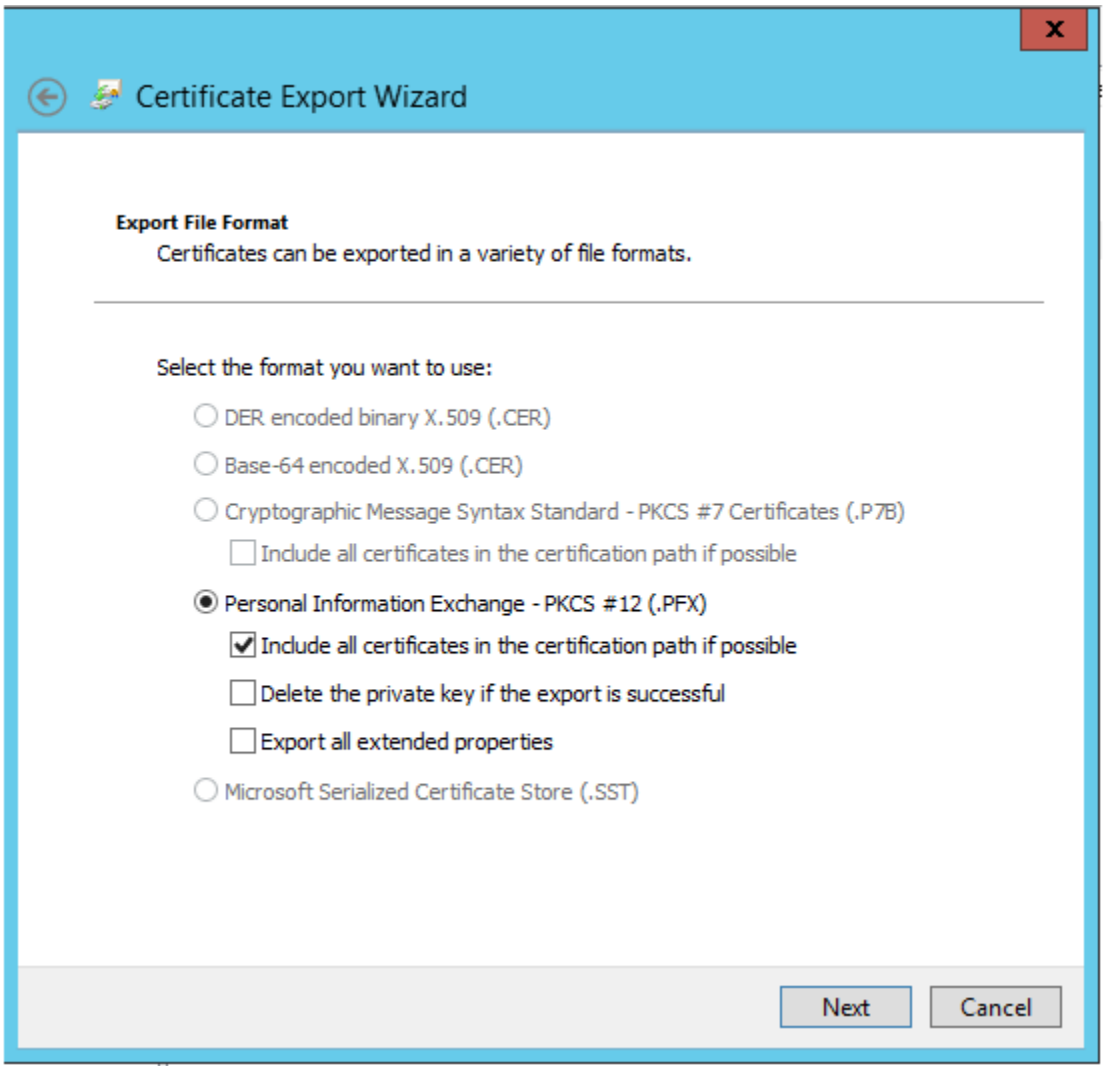
From the certificate snap-in, right click on our new certificate and select All Tasks > Export.



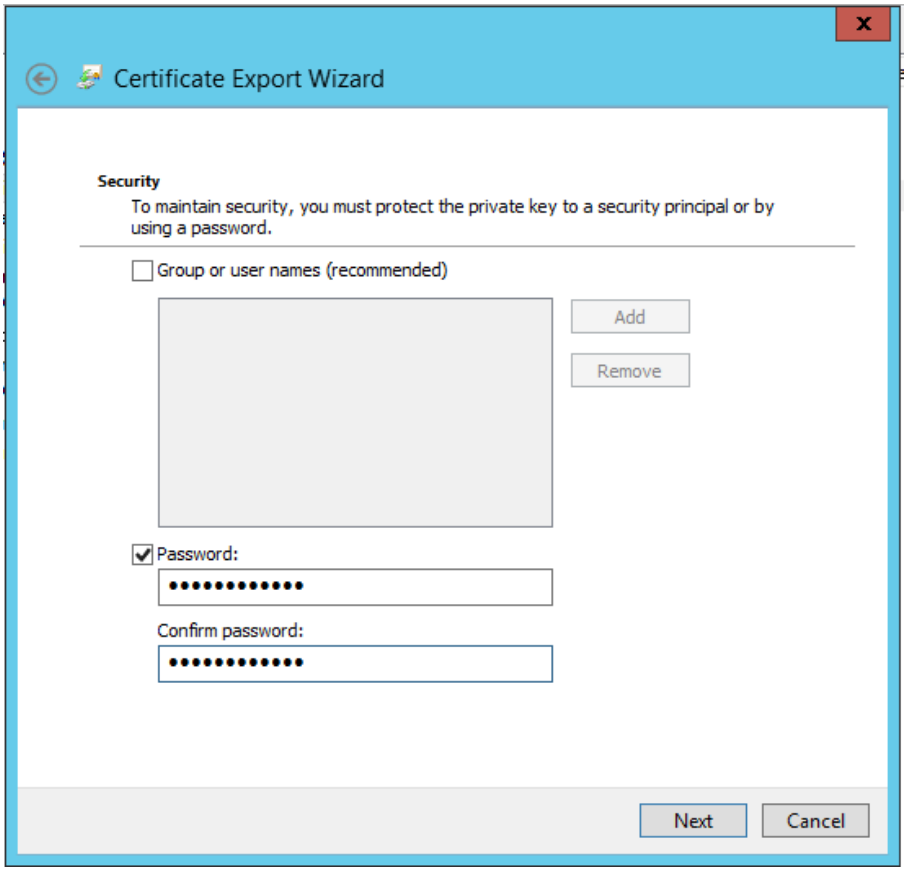
This brings up the Certificate Export Wizard.

Click next and on the next window, select Yes, export the private key and click next.

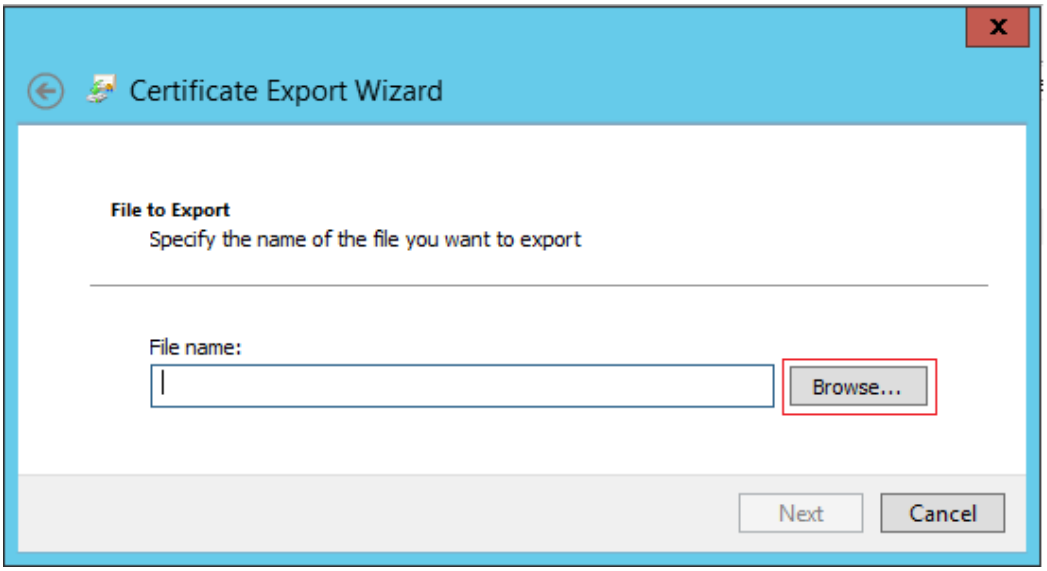
Ensure the Personal Information Exchange -PKCS #12 (.PFX) option is selected and the box the first box from the list of options is checked as well. Click next.



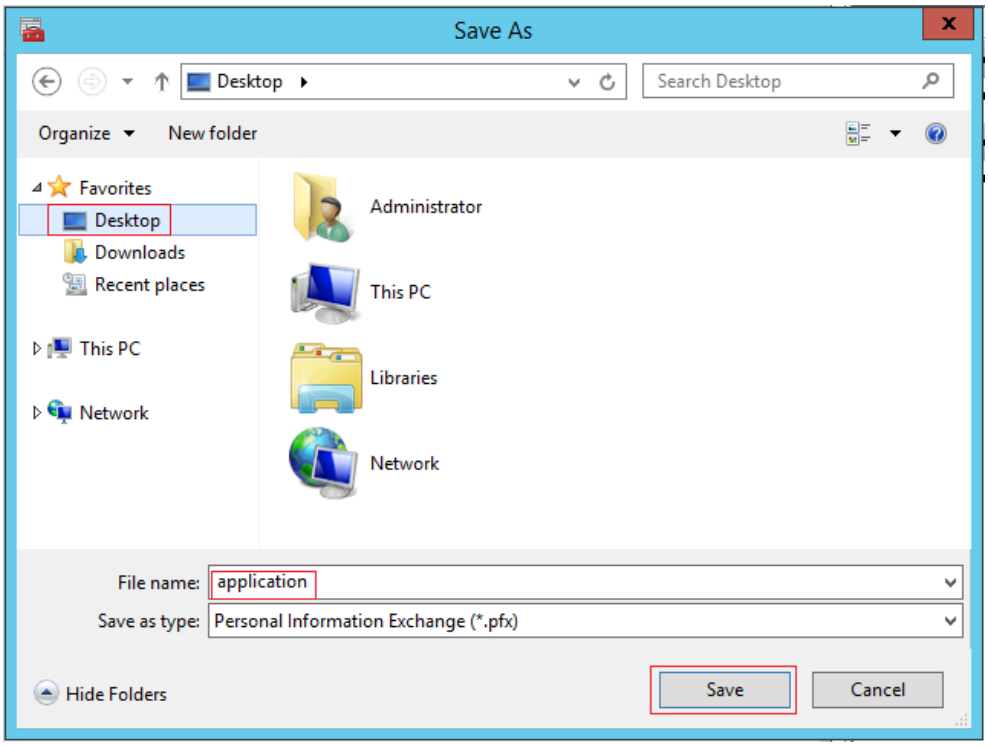
On the next screen, check the box for using a password. Type in any password you want.



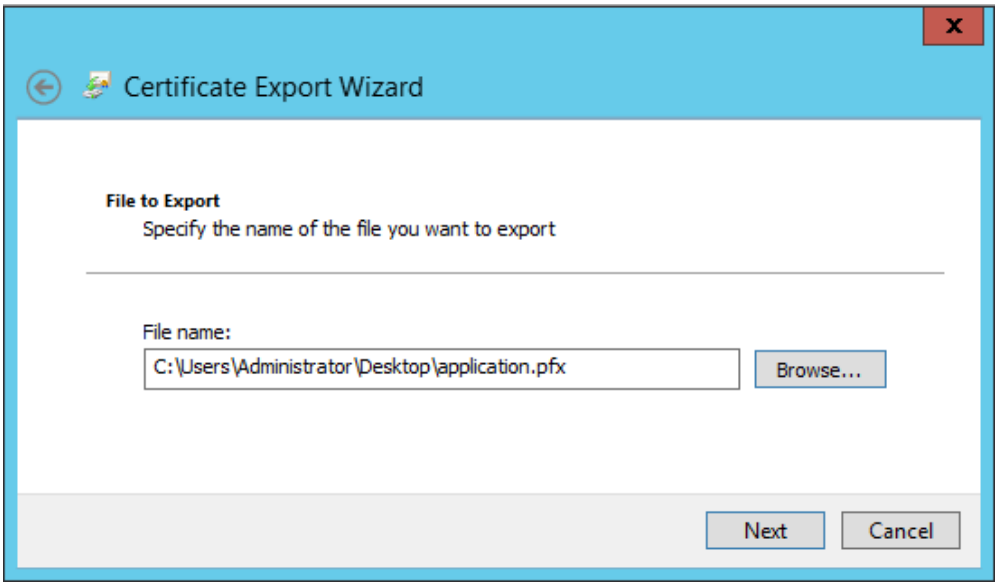
On the next screen, click the browse button.



On the next screen, select the Desktop. Choose a file name to use for the export. In this example, I have named the exported file, application and saved it to the servers Desktop.

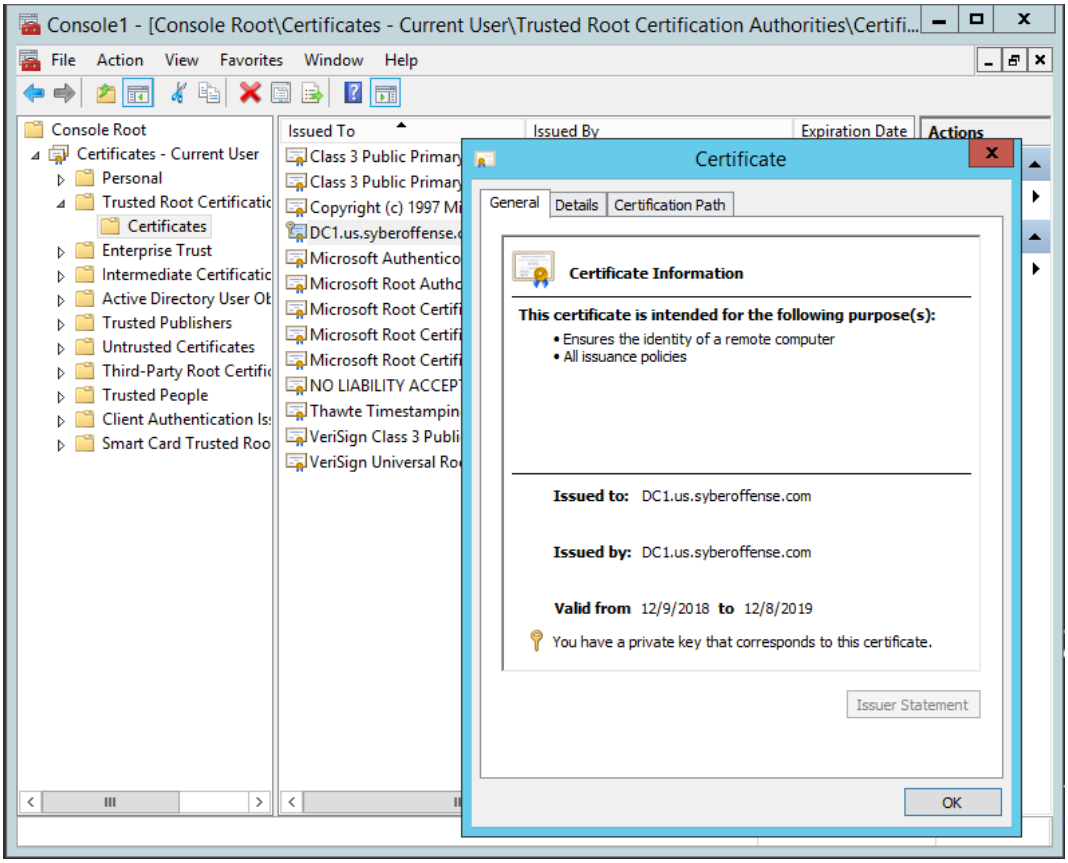


On the next screen, click the Next button.



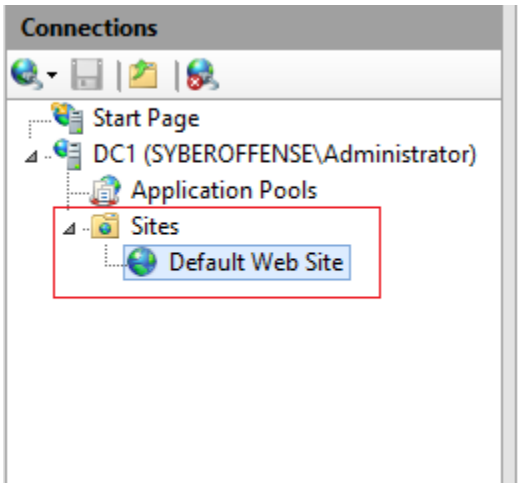
On the Complete the Certificate Export Wizard screen, click finish and then click OK.

Minimize any open windows and find the certificate saved the desktop. To view the certificate, bring up your certificate snap-in, find the certificate and x2 click to view.

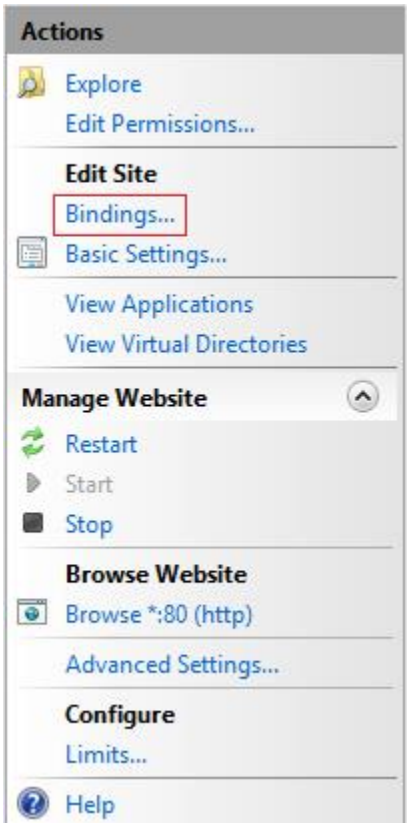


Bind a Certificate to a site

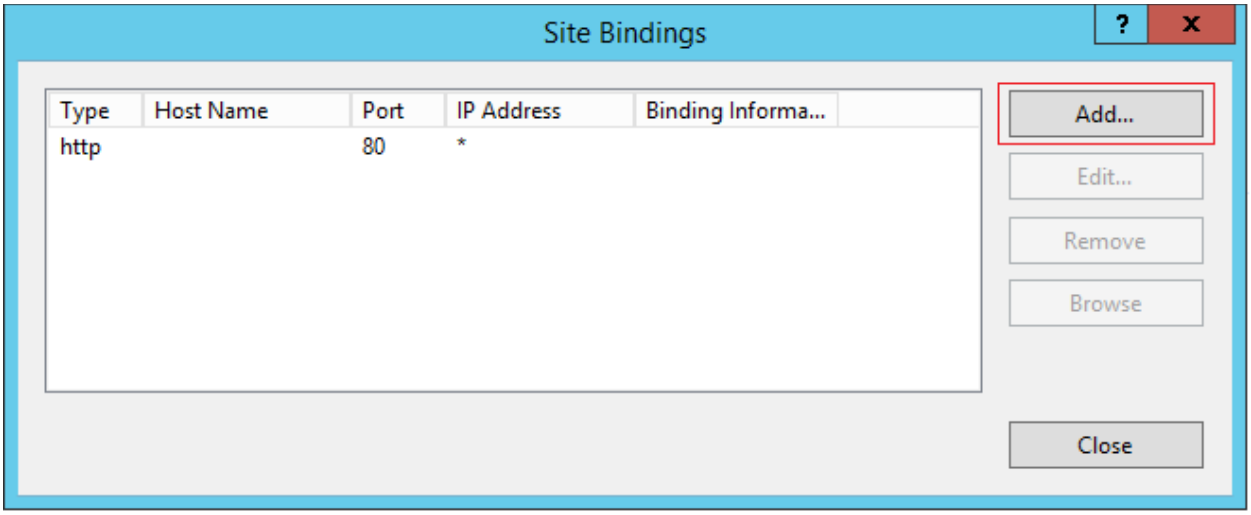
Bring up your IIS management console. In the left windows pane, click on Sites and then click on your Default Web Site.



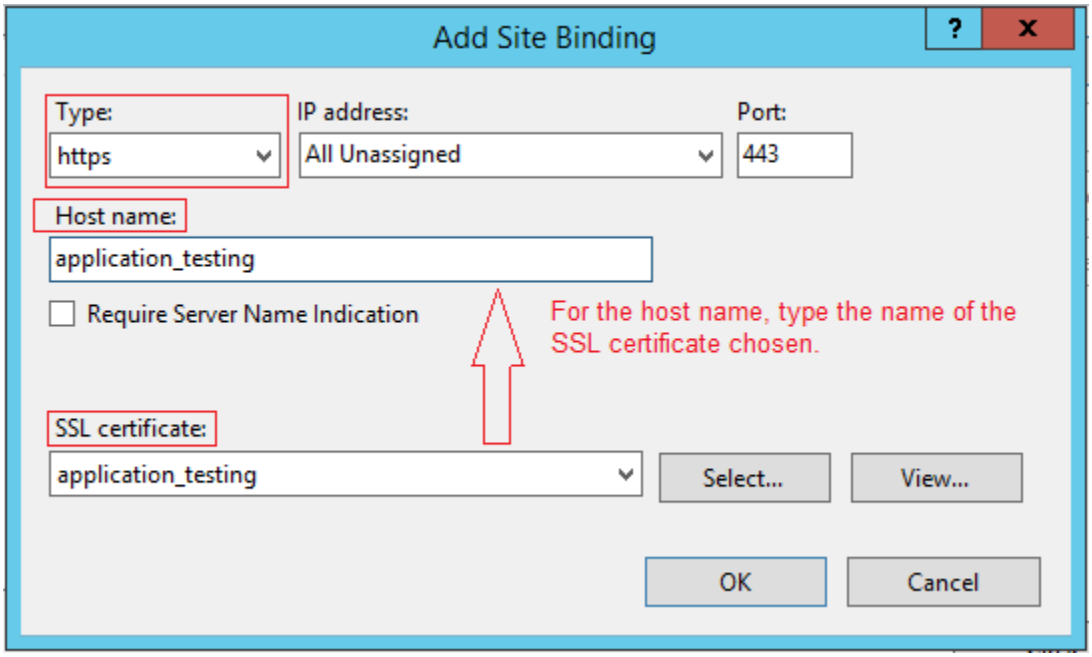
Ove on the far-right Action window pane, click on bindings.



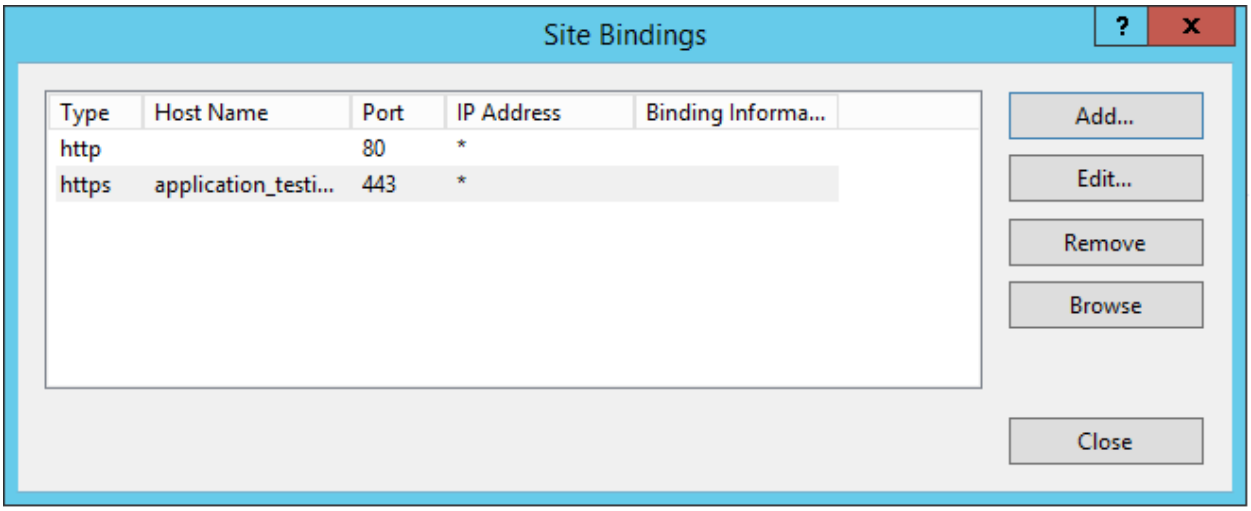
On the next window, click the Add button.



On the next screen, under the Type option, pull down the windows and select https. Under the SSL certificate, pull down the window and select the name of the certificate we created. Under Host names, type the name of the SSL certificate you selected. Click OK.



On the next window, click close.

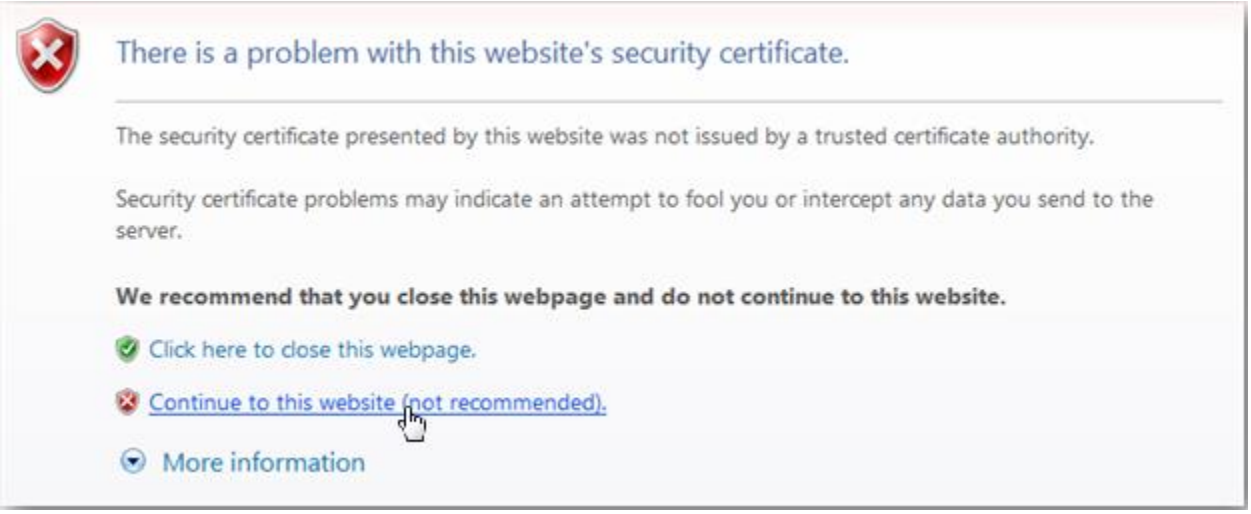


Users can now use HTTPS to browse the default web site.

Summary

In this lab, we learned how to generate a self-signing certificate using IIS removing the need of having to create a PKI infrastructure and manage a certificate server. For creating a temporary certificate for an application or website development, this a quick and easy way to create an SSL certificate for an internal trusted source. The trick is to ensure the certificate is placed inside the trusted root certificate folder on the web server and the user can import the certificate to their local machine using their browser.

With a self-signing certificate from an untrusted source, the user will see the following security warnings in most browsers.

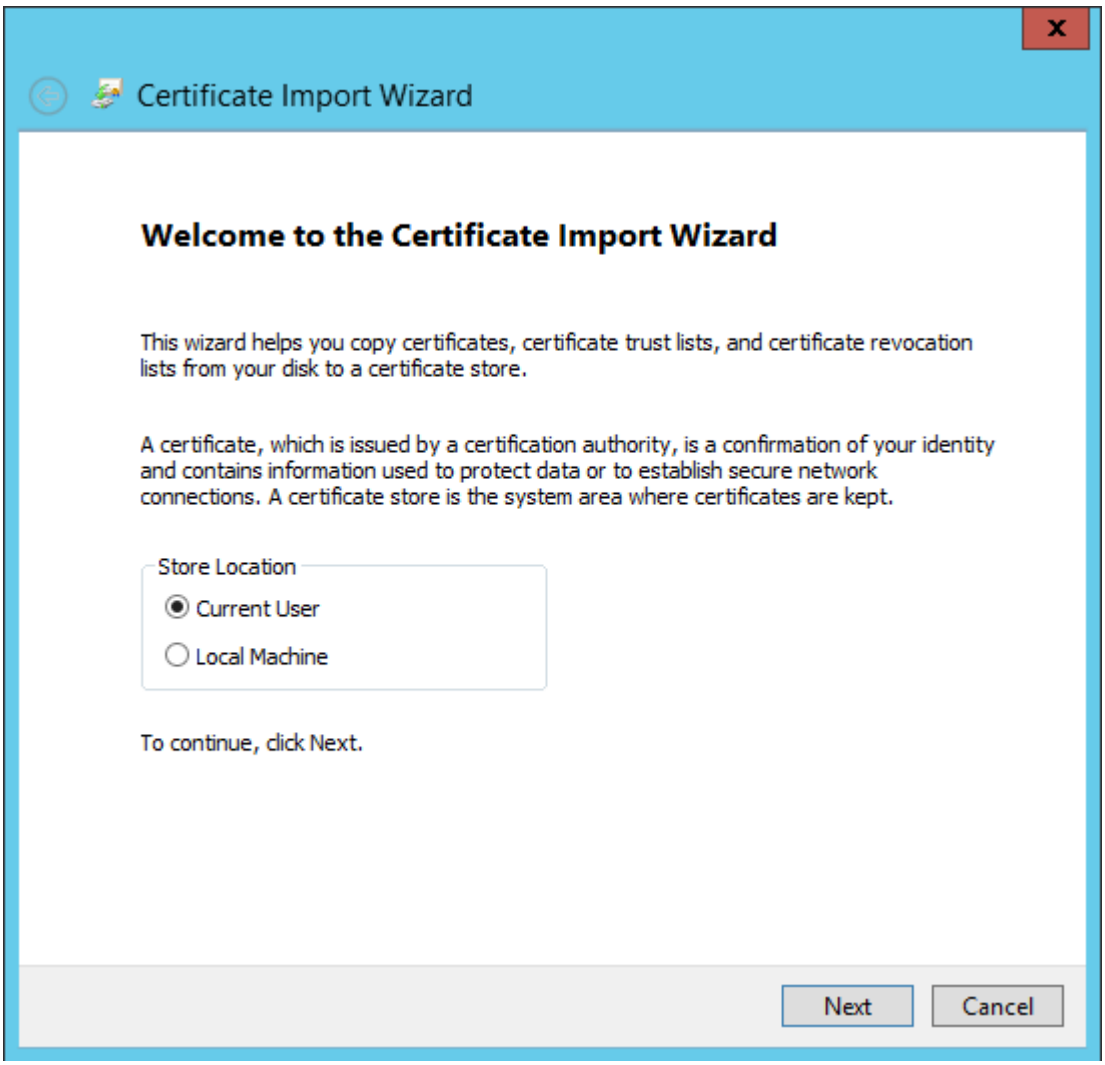


The user can proceed onto the website or import the self-signed certificate to their local machine.

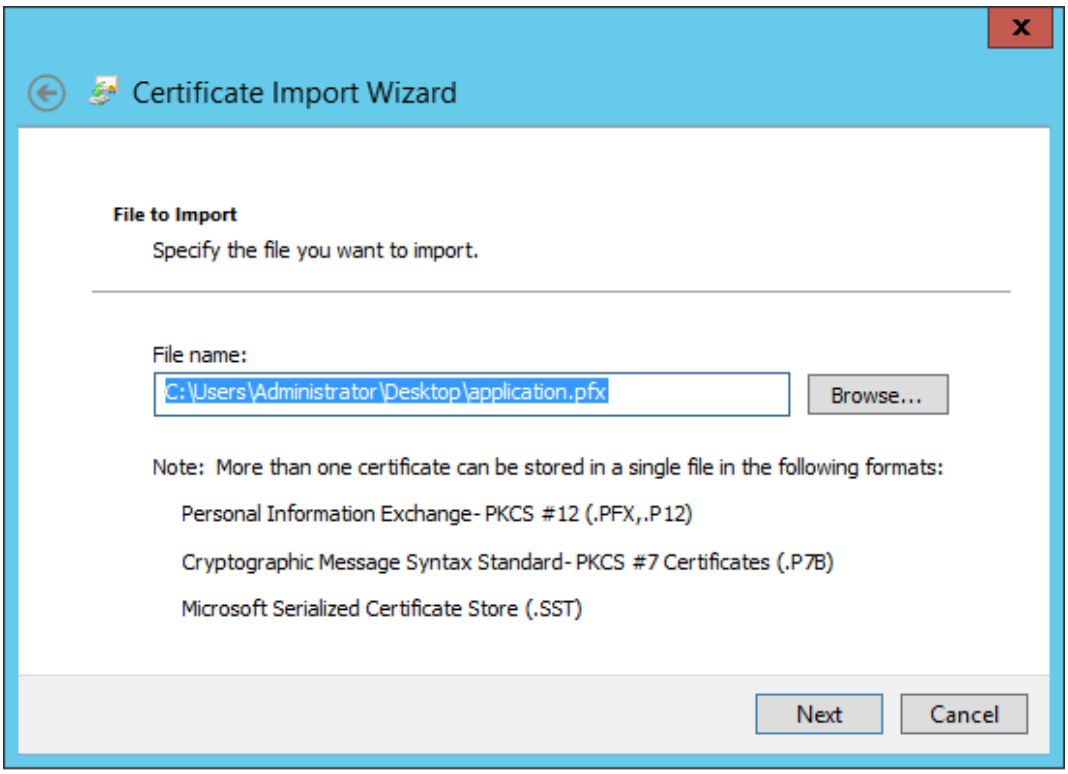
Importing the self-signed certificate to the client machine

Copy the certificate which was exported from the server (the PFX file) to the client's machine or ensure it is available from a network path.

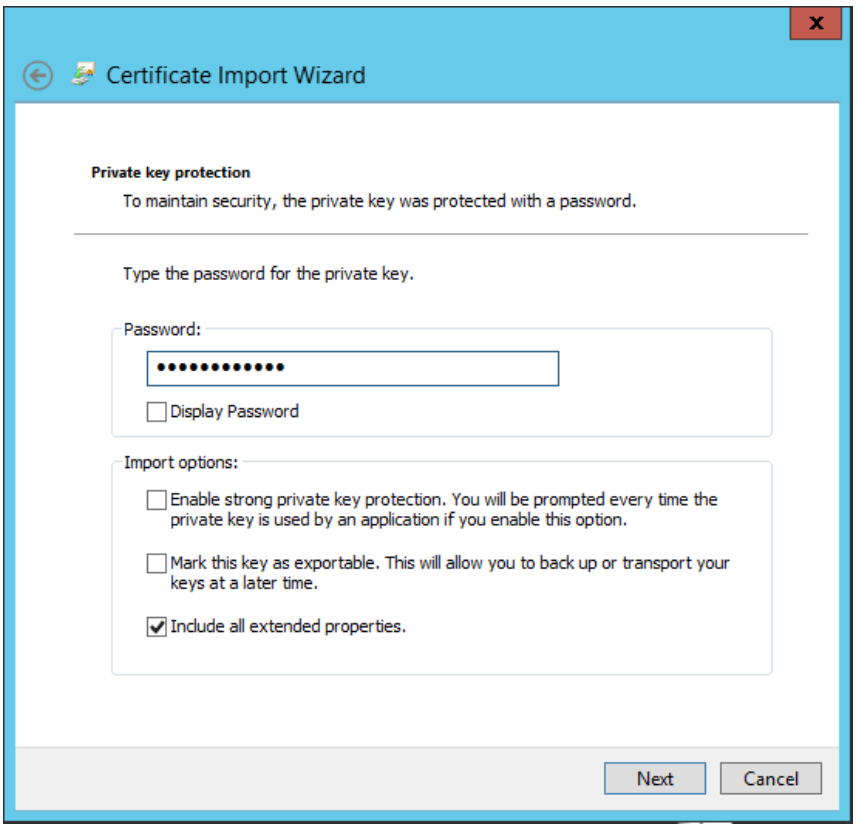
From the client's machine, right click on the certificate and select install. On the first screen, select Current User.



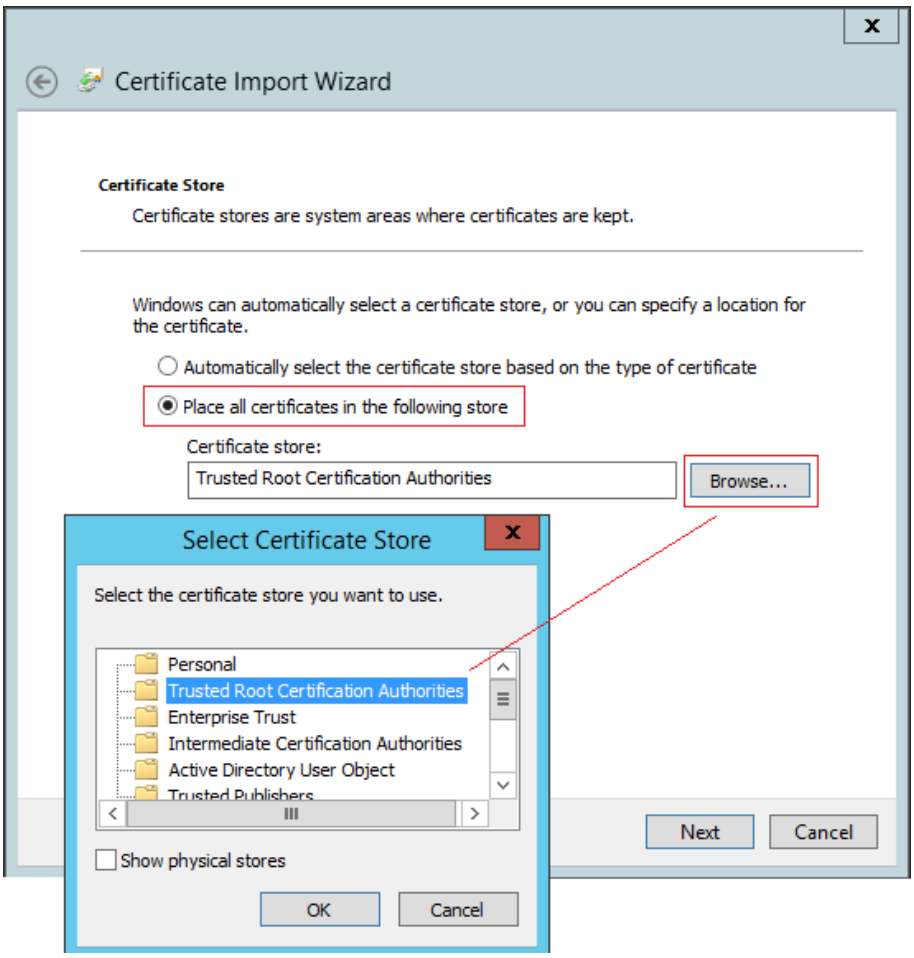
On the next screen, verify the certificate you are importing.



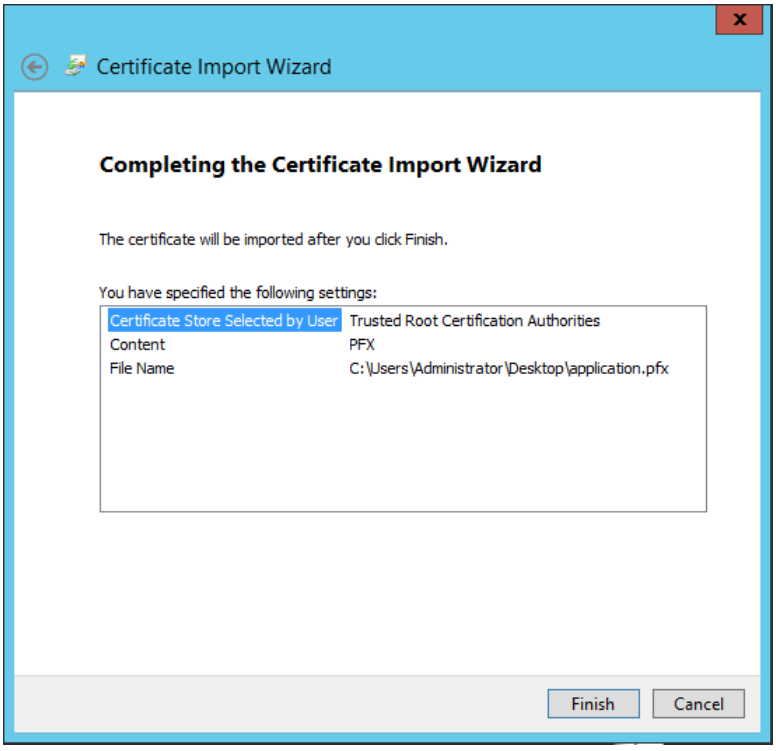
Type in the certificate's password we create at the beginning of the lab.



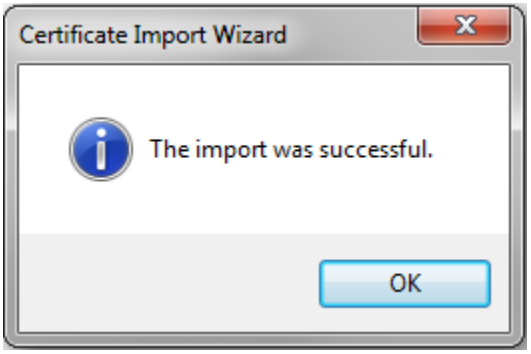
On the next screen, select the second radio button and browse to the local certificate store and select the Trusted Root Certification Authorities container.



On the next screen, click Finish to import the certificate.



Say ok to the confirmation message.



One this is done, users should be able to browse to an HTTPS site which uses these certificates and receive no warnings or prompts.

Important Note: Users should *never* install a security certificate from an unknown source. In practice, you should only install a certificate locally if you generated it. No legitimate website would require you to perform these steps.