SENG2250/6250 System and Network Security

School of Electrical Engineering and Computing

Semester 2, 2020

Lab 9: Secure Socket Layer

Objectives

1) Create an SSL protected web site.
2) Inspect SSL handshake traffic.

Part 1 Lab Environment

1. This lab should be conducted under the **Windows 10** VM. To access your virtual lab:

   https://cybersec-vra.newcastle.edu.au/vcac/org/cybersec

   Username: Student

   Password: $tud3nt

2. This lab requires XCA, XAMPP, Firefox, and Wireshark.

Part 2 Exercises 1: Create an SSL Protected Web Site

Task 1: Creating a Certificate Authority (CA) using XCA

In this task, we will first create a CA. The role of CA is to issue public-key certificates to end entities (e.g., our web server). In other words, the CA digitally signs a public key of a web server and embeds it into the server's certificate. The CA uses its private key for signing. In practice, the CA certificate manager (a certificate server) should be installed on a secure machine (potentially disconnected from the network); the CA's private key should be kept highly secure.

The CA also issues a self-signed public-key certificate, whereby the CA digitally signs its own public key. This certificate is distributed, in a secure way (the integrity of the CA's certificate must be protected), to all users who use a certificate issued by the CA to verify the authenticity of the certificate holder (e.g., a web server).

1. Open XCA. (C:\Program Files (x86)\xca\xca.exe)
   Create a new database to store your keys and certificates. Select "File" and "New Database", name it as "mydatabase". You will be asked for a password, but it is not necessary for this lab. You can simply click "Ok".
2. Click the "New Certificate" button to start the X.509 certificate creation procedure.

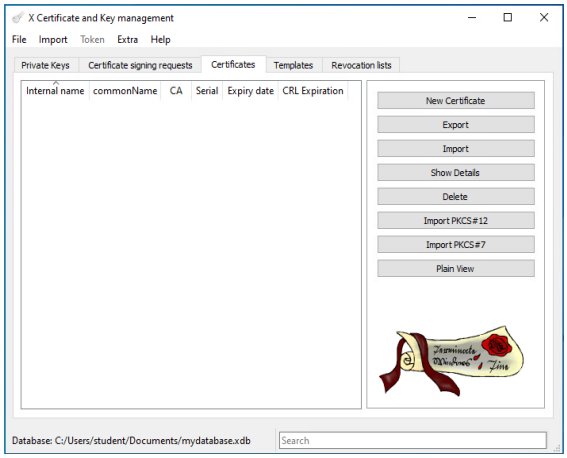

Figure 1. Starting new certificate generation.

3. Select CA template for this certificate.
4. Select the "Subject" tab and fill the fields, i.e., "Internal name", "Country name", etc. Please refer to Figure 2 as an example. Click "Generate a new key" button to generate a new private key (2048-bit RSA). This will be the signing key of the CA.
5. Select the "Extensions" tab and set the "Type" of the certificate "Certificate Authority", then click "OK" to finish the certificate creation procedure.

   You can check the details of the created CA certificate by double-clicking on it in the main XCA window under the "Certificate" tab.

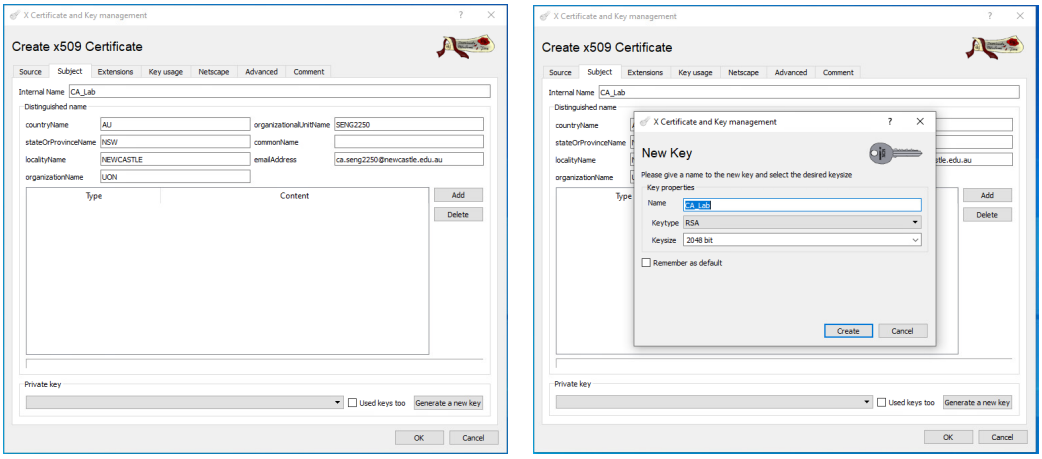

Figure 2. Create CA certificate.

Task 2: Create a Web Server Certificate

In this task we create a public-key certificate for a web server. This certificate will be digitally signed by the previously created Certification Authority (CA).

1. Click the "New Certificate" button in the main XCA window. Select the "Source" tab and configure the web server certificate properties as follows: Check "Use this Certificate for signing" and set it to the name of the CA created in the previous task. Choose "HTTPS server template" from the list of available templates.
2. Select the "Subject" tab and fill the fields properly. Please refer to Figure 3 as an example. It is important to note that fill the commonName as "localhost". This will be the IP address (or the corresponding URL) of your web server. Finally, click "Generate a new key" button to generate a new private key (2048-bit RSA).

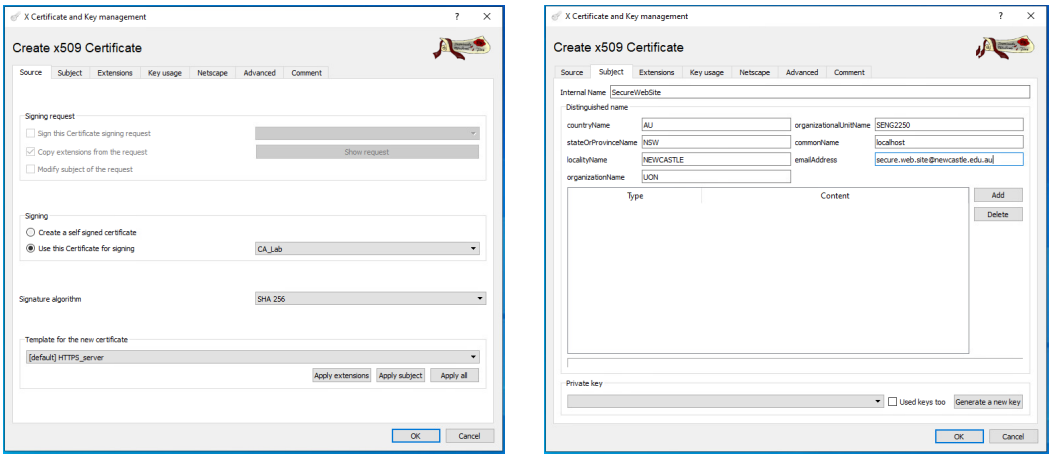

Figure 3. Create web server certificate.

3. Under the "Extensions" tab, set the type to "End Entity", then click "OK" to finish the certificate creation procedure.

Task 3: Create a Client Certificate

In this task we create a public-key certificate for a client. This certificate will be also digitally signed by the previously created Certification Authority (CA).

1. Click the "New Certificate" button in the main XCA window. Select the "Source" tab and configure the client certificate properties as follows: Check "Use this Certificate for signing" and set it to the name of the CA created in the previous task. Choose "HTTPS client template" from the list of available templates.
2. Select the "Subject" tab and fill the fields properly. Please refer to Figure 4 as an example. Then, click "Generate a new key" button to generate a new private key (2048-bit RSA).

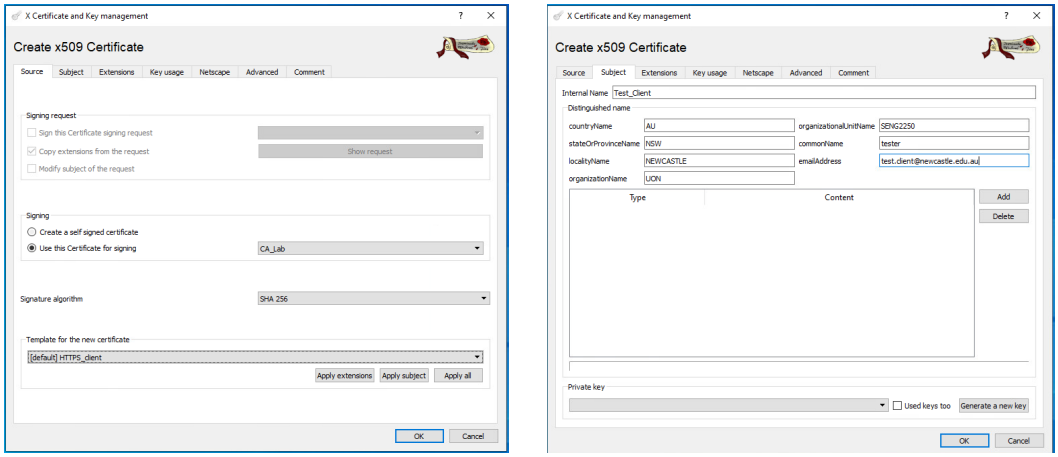

Figure 4. Create client certificate.

3. Under the "Extensions" tab, set the type to "End Entity", then click "OK" to finish the certificate creation procedure.

Task 4: Export Certificates

In this task, we will export the certificates created in the previous tasks, that is, the CA public-key certificate (without the private key), the web server public-key certificate including its private key, as well as the client public-key certificate with its private key.

1. Open the "Certificates" tab in the main XCA window. Select the certificate that belongs to the CA and click "Export". Select a destination (e.g., Documents) and filename where you want to store the certificate and click "OK". Note that you have to change the default path due to the permission issues.
2. Repeat the previous step but now export the public-key certificate of the web server and a client certificate.
3. Finally, we export the private key of web server and of client.
    a. For web server: click the "Private Keys" tab from the XCA main window. Select web server private key and click "Export". Choose a destination (e.g., Documents) and filename where you want to store the private key. Finally, click OK to export the web server private key. (Figure 5)

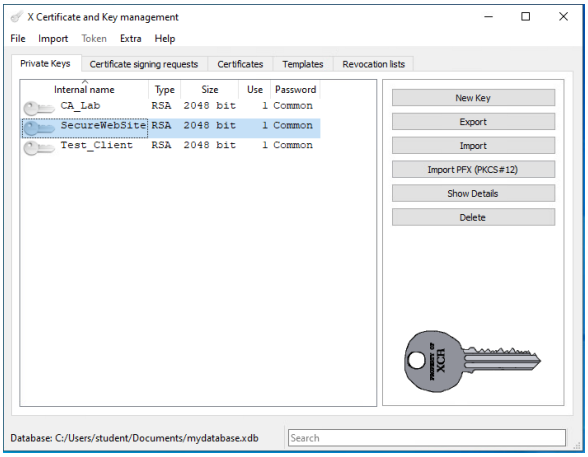

Figure 5. Export web server private key.

b. For client: select the client certificate under the “Certificates” tab, and click “Export”. Select a destination (e.g., Documents) and filename where you want to store the private key. Then, change the “Export Format” to “PKCS #12 (.p12)” (Figure 6). It will be imported to the web browser. You will be asked to create a password, you may simply click “OK” to skip it.

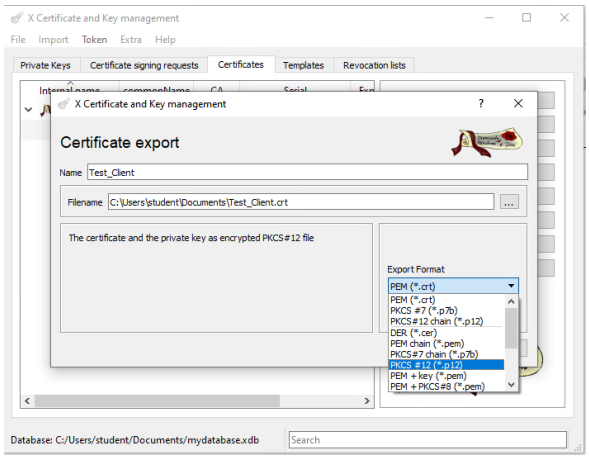

Figure 6. Export client private key.

You should have created files showed as in Figure 7.

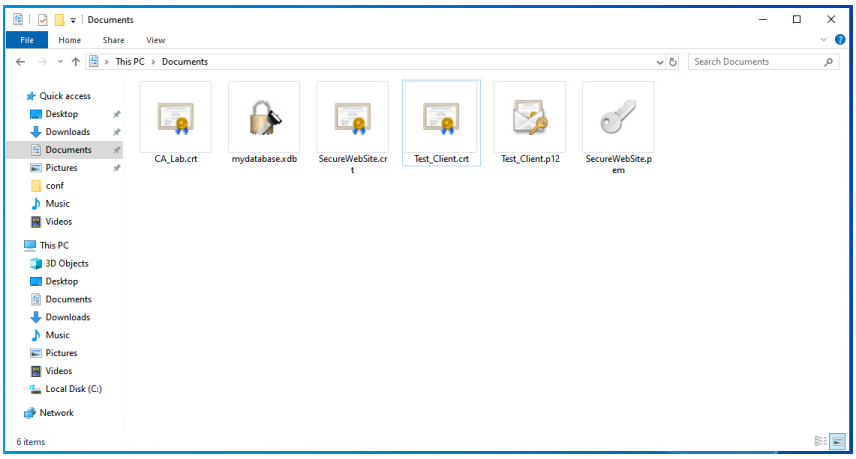

Figure 7. Created files.

Task 5: Configure Apache Web Server

1. To configure the Apache Web Server, we will need the CA_Lab.crt and SecureWebSite.crt public- key certificates as well as the private key SecureWebSite.pem. Copy these certificates and keys to the conf folder of the Apache Web Server (e.g. C:\xampp\apache\conf).
2. Edit httpd.conf (e.g. found within "C:\xampp\apache\conf" folder) and ensure the following lines are uncommented:

   LoadModule ssl_module modules/mod_ssl.so

   Include conf/extra/httpd-ssl.conf

3. Edit httpd-ssl.conf (e.g. found within "C:\xampp\apache\conf\extra" folder) and verify that the following holds (should be uncommented and the value is correct):
   a. SSLCertificateFile ''C:/xampp/apache/conf/SecureWebSite.crt''
   b. SSLCertificateKeyFile ''C:/xampp/apache/conf/SecureWebSite.pem''
   c. SSLCertificateChainFile ''C:/xampp/apache/conf/CA_Lab.crt''
   d. SSLCACertificateFile ''C:/xampp/apache/conf/CA_Lab.crt''
   e. SSLVerifyClient require
   f. SSLVerifyDepth 2
4. Start XAMPP Control Panel. Run the program: "C:\xampp\xampp-control.exe".
5. Start/Restart the Apache server.

Task 6: Test Your Configuration

In this task we will install a client public-key certificate in the browser and test our configuration. To accomplish this goal in a Firefox browser, we can place the client public-key certificate in the "Your Certificates" directory. Here are the steps to install the client public-key certificate

1. Click the "Firefox menu" on the browser toolbar.
2. Select "Options".
3. Click "Privacy & Security", on the left of the page.
4. Go to "View Certificates" to manage your SSL certificates and settings.
5. Click on the "Your Certificates" tab and select "Import...".
6. Select the client's .p12 file (e.g., Test_Client.p12) and click "Open".
7. You will be asked to enter the password created while exporting client public-key certificate to pkcs12. If you didn't create a password, leave it blank.
8. Click "Ok".
9. Open a new tab in Firefox and enter the following address in the address bar:

   https://localhost/dashboard

   Do you get any warning message? Why?

10. We would like to eliminate this warning message. What can we do in this regard?
11. Recall that the CA has digitally signed the web server public-key certificate. So if our web browser would have access to the CA certificate (i.e., if it would trust this certificate), the web browser could successfully verify the digital signature in the web server certificate and would not report any warning messages.

12. To accomplish this goal in a Firefox browser, we can place the CA's certificate in the "Authorities" directory. Here are the steps to install the CA's certificate:
    a. Click the "Firefox menu" on the browser toolbar.
    b. Select "Options".
    c. Click "Privacy & Security", on the left of the page.
    d. Click "View Certificates".
    e. Click on the "Authorities" tab and select "Import...".
    f. Select the CA's .crt file (e.g., CA_Lab.crt) and click "Open".
    g. Tick both check boxes.
    h. Click "Ok".

13. Now that you have installed the CA certificate, try again to access:

https://localhost/dashboard

Do you get any warning message? Why?

Part 3 Exercises 2: Inspect SSL Handshake Traffic

An important part of SSL is the initial handshake that establishes a secure connection. This exercise aims to inspect the messages of different phases during an SSL handshake.

Task 1: Preparation

1. Open the file (located on your Desktop) "SENG2250_Lab_TLS.pcapng".
2. Locate the SSL handshake flow. Type "ip.addr == 134.148.4.137 && ssl.handshake" in the "Apply a display filter …" textbox and apply.

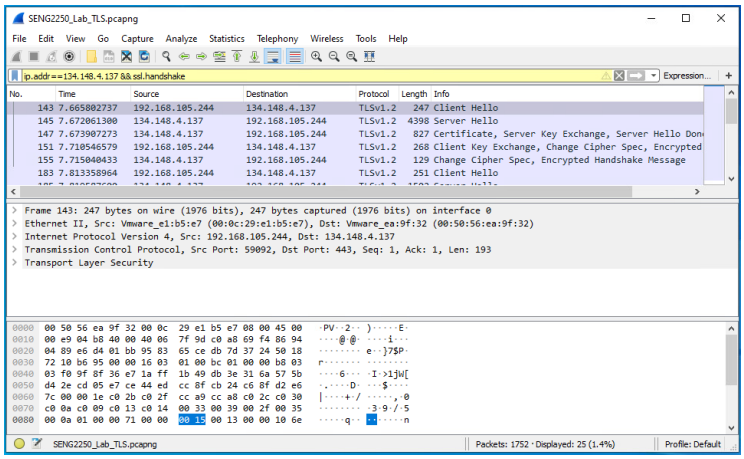

Figure 8. SSL Handshake in Wireshark

Task 2: Inspect Client Hello Message

Find the Client Hello message. Inspect "Transport Layer Security" and answer the following questions:

1. How many cipher suites are supported by the client?
2. What is the cipher suite with the code "**0x0033**"?
3. A cipher suite will indicate:

   key exchange protocol, **signature algorithm**, **block cipher scheme**, **block cipher key size**, **operation mode** and **hash function**.

   Explain the meaning of cipher suite "**0x0033**".
4. What is the length of "Session ID"? Why?
5. Look up the "Extensions" and find how many signature hash algorithms are supported by the client.

Task 3: Inspect Server Hello Message

Find the Server Hello message. Inspect "Transport Layer Security" and answer the following questions:

1. Is there any value assigned to the "Session ID"? Who creates it?
2. Which "Cipher Suite" is selected (agreed)? Explain the selected cipher suite.
3. Is there any compression applied?

Task 4: Inspect Certificate, Server Key Exchange, Server Hello Done message

Find the Certificate, Server Key Exchange, Server Hello Done message. Inspect "Transport Layer Security" and answer the following questions:

1. What is the server's identity?
2. How many certificates were transferred? Why do we need all the certificates?
3. Which key exchange protocol was used for the handshake?
4. What is the size of "public key" used in key exchange server parameters?
5. Which signature algorithm was used for key exchange
6. Is there any information transferred for Server Hello Done?

Task 5: Inspect Client Key Exchange Message

Find the Client Key Exchange message. Inspect "Transport Layer Security" and answer the following questions:

1. Comparing it with the Server Key Exchange message, what are differences between "Diffie-Hellman Client Params" and "Diffie-Hellman Server Params"?
2. What is the content and the length of Change Cipher Spec Message payload? Explain.
3. What content is provided in the Encrypted Handshake Message? As the message was encrypted, we cannot read it. What was the encryption key used here?

Supplementary Task

Use Wireshark to capture the traffic while accessing the secure web site (created in Exercise 1). Inspect the SSL messages as the above.

Wireshark tutorial:

https://www.wireshark.org/docs/wsug_html_chunked/ChapterCapture.html