Lab Exercise – SSL

**Objective**

To observe SSL/TLS (Secure Sockets Layer / Transport Layer Security) in action. SSL/TLS is used to secure TCP connections, and it is widely used as part of the secure web: HTTPS is SSL over HTTP. SSL/TLS is covered in §8.9.3 of your text. Review that section before doing this lab.

**Step 2: Inspect the Trace**

Now we are ready to look at the details of some “SSL” messages. There are several different versions of SSL and TLS that are in widespread use, including SSL version 2, SSL version 3, and TLS version 1; TLS is the open standard version of the protocol and TLS 1.0 comes after SSL 3.0. Following common practice, we will informally refer to all of them as “SSL”. A complication for this lab is that they all behave slightly differently. To minimize variation, we asked you to gather TLS 1.0 traffic as part of the steps above.

*To begin, enter and apply a display filter of “*ssl*”.* This filter will help to simplify the display by showing only SSL and TLS messages. It will exclude other TCP segments that are part of the trace, such as Acks and connection open/close. Your display should be similar to the one shown in our figure.



Figure 3: Trace of “SSL” traffic showing the details of the SSL header

*Select a TLS message somewhere in the middle of your trace for which the Info reads “Application Data”, and expand its Secure Sockets Layer block (by using the “+” expander or icon).* Application Data is a generic TLS message carrying contents for the application, such as the web page. It is a good place for us to start looking at TLS messages.

*Look for the following protocol blocks and fields in the message:*

• The lower layer protocol blocks are TCP and IP because SSL runs on top of TCP/IP.

• The SSL layer contains a “TLS Record Layer”. This is the foundational sublayer for TLS. All messages contain records. Expand this block to see its details.

• Each record starts with a Content Type field. This tells us what is in the contents of the record.

• Then comes a Version identifier. It will be a constant value for the SSL connection.

• It is followed by a Length field giving the length of the record.

• Last comes the contents of the record. Application Data records are sent after SSL has secured the connection, so the contents will show up as encrypted data. To see within this block, we could configure Wireshark with the decryption key. This is possible, but outside of our scope.

Note that, unlike other protocols we will see such as DNS, there may be multiple records in a single message. Each record will show up as its own block. Look at the Info column, and you will see messages with more than one block.

*Answer the following questions to show your understanding of SSL records:*

1. *What is the Content-Type for a record containing “Application Data”?*

Content Type: Application Data (23)

2. *What version constant is used in your trace, and which version of TLS does it represent?*

Version: TLS 1.0 (0x0301)

3. *Does the Length cover the Record Layer header as well as payload, or only the payload?*

Only the Payload

**Turn-in**: Hand in your answers to the above questions.

**Step 3: The SSL Handshake**

An important part of SSL is the initial handshake that establishes a secure connection. The handshake proceeds in several phases. There are slight differences for different versions of TLS and depending on the encryption scheme that is in use. The usual outline for a brand new connection is:

a. Client (the browser) and Server (the web server) both send their Hellos

b. Server sends its certificate to Client to authenticate (and optionally asks for Client Certificate)

c. Client sends keying information and signals a switch to encrypted data.

d. Server signals a switch to encrypted data.

e. Both Client and Server send encrypted data.

f. An Alert is used to tell the other party that the connection is closing.

Note that there is also a mechanism to resume sessions for repeat connections between the same client and server to skip most of steps b and c. However, we will not study session resumption.

**Overall Handshake**

*To get a sense of the overall SSL connection behavior, draw a timeline showing and naming the SSL messages sent between the client and the server.* As usual, draw vertical lines to represent the client and the server, with time running down the page. To work out the names of the SSL messages you should include, use the description given in the Info column of your Wireshark display. You should cover the entire connection, but elide most of the encrypted SSL messages in the middle of the connection (Application Data) to keep the figure simple. There is no need for you to understand the details at this stage; our next step is to look at each of the phases of the connection in more detail.

Sever

Client

**Info** = Sever Hello

**Info** = Client Hello

**Info** = Client Key Key Exchange,

Change Cipher Spec,

Encrypted Handshake Message

**Info** = Certificate

**Info** = Change Cipher Spec,

Encrypted Handshake Message

**Info** = Application Data

**Info** = Application Data

**Info** = Application Data

**Info** = Application Data

**Info** = Application Data

**Info** = Encrypted Alert

**Hello Messages**

*Find and inspect the details of the Client Hello and Server Hello messages, including expanding the Hand-shake protocol block within the TLS Record.* For these initial messages, an encryption scheme is not yet established so the contents of the record are visible to us. They contain details of the secure connection setup in a Handshake protocol format.

*Answer the following questions:*

1. *How long in bytes is the random data in the Hellos?* Both the Client and Server include this random data (a nonce) to allow the establishment of session keys.

Random data = 32 bytes

2. *How long in bytes is the session identifier sent by the server?* This identifier allows later resumption of the session with an abbreviated handshake when both the client and server indicate the same value. In our case, the client likely sent no session ID as there was nothing to resume.

Session ID Length: 32

3. *What Cipher method is chosen by the Server? Give its name and value.* The Client will list the different cipher methods it supports, and the Server will pick one of these methods to use.

Cipher Suite: TLS\_RSA\_WITH\_RC4\_128\_SHA (0x0005)

**Certificate Messages**

*Next, find and inspect the details of the Certificate message, including expanding the Handshake proto-col block within the TLS Record.* As with the Hellos, the contents of the Certificate message are visible because an encryption scheme is not yet established. It should come after the Hello messages.

*Answer the following questions:*

4. *Who sends the Certificate, the client, the server, or both?* A certificate is sent by one party to let the other party authenticate that it is who it claims to be. Based on this usage, you should be able to guess who sends the certificate and check the messages in your trace.

Only the Server sends the Certificate.

A Certificate message will contain one or more certificates, as needed for one party to verify the identity of the other party from its roots of trust certificates. You can inspect those certificates in your browser.

**Client Key Exchange and Change Cipher Messages**

*Find and inspect the details of the Client Key Exchange and Change Cipher messages, expanding their various details.* The key exchange message is sent to pass keying information so that both sides will have the same secret session key. The change cipher message signal a switch to a new encryption scheme to the other party. This means that it is the last unencrypted message sent by the party.

*Answer the following questions:*

5. *At the Record Layer, what Content-Type values are used to indicate each of these messages? Say whether the values are the same or different than that used for the Hello and Certificate messages.* Note that this question is asking you to look at the Record Layer and not an inner Hand-shake Protocol.

Content Type: Change Cipher Spec (20)

It is different from what is used for Hello and Certificate messages.

6. *Who sends the Change Cipher Spec message, the client, the server, or both?*

Both.

7. *What are the contents carried inside the Change Cipher Spec message?* Look past the Content-Type and other headers to see the message itself.

Content Type: Change Cipher Spec (20)

**Alert Message**

*Finally, find and inspect the details of an Alert message at the end of the trace.* The Alert message is sent to signal a condition, such as notification that one party is closing the connection. You should find an Alert after the Application Data messages that make up the secure web fetch.

*Answer the following questions:*

8. *At the Record Layer, what Content-Type value is used to signal an alert?*

Content Type: Alert (21)

9. *Tell us whether the contents of the alert are encrypted or sent in the clear?* To check this, see whether you can read the contents of the alert to see what kind of alert has been sent.

Alert Message: Encrypted Alert

It is encrypted.

**Turn-in**: Hand in your answers to the above questions.