Lab Exercise – SSL/TLS

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.

Step 1: Open a Trace

Proceed as follows to capture a trace of SSL traffic; alternatively, you may use a supplied trace. The easiest way for us to produce SSL traffic is to fetch web pages with HTTPS. Any URL with HTTPS will do, e.g., https://www.google.com.

1. *Close all unnecessary browser tabs and windows.*
2. *Launch Wireshark and start a capture with a filter of "*`tcp port 443`*".* We use this filter because there is no shorthand for SSL, but is normally carried on port 443 in case of secure pages.
3. *Open Wireshark trace http://scisweb.ulster.ac.uk/~kevin/com320/labs/wireshark/trace-ssl.pcap*

Step 2: Inspect the Trace

Now we are ready to look at the details of some "SSL" messages. *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.

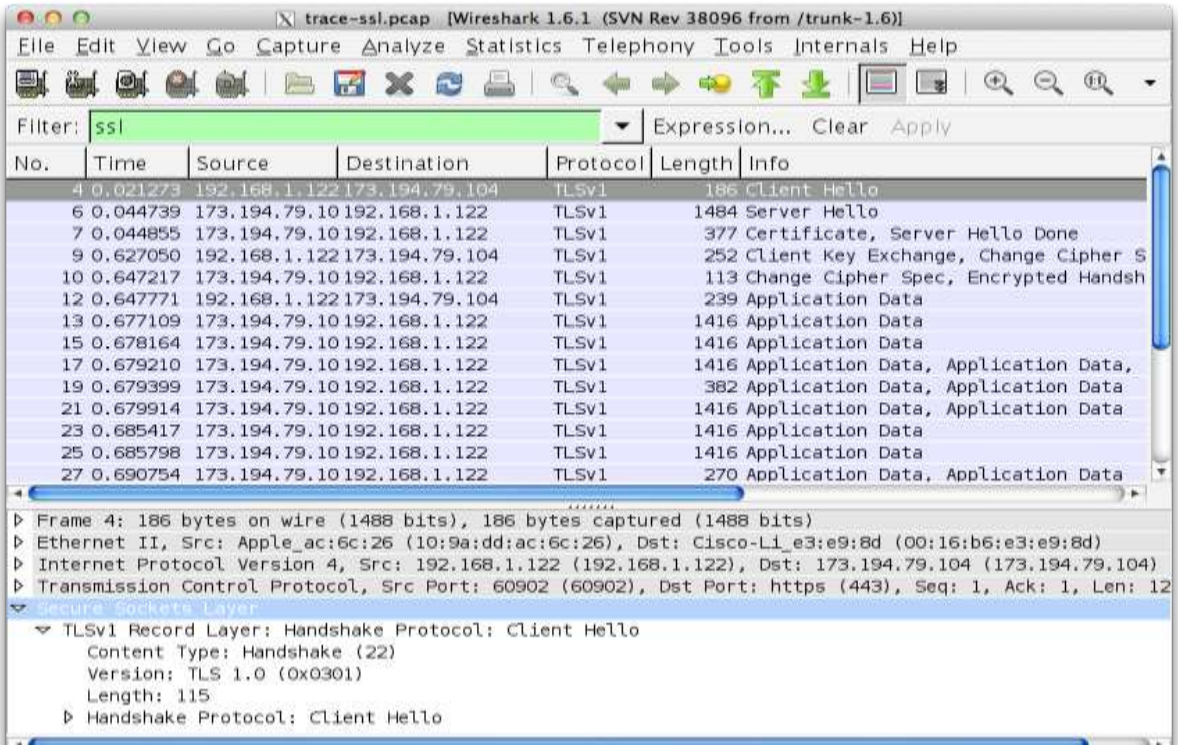

Figure 3: Trace of "SSL" traffic showing the details of the SSL header

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.

Questions (Answers are below)

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

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

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

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

Answers

1. A Content-Type value of 23 indicates "Application Data".

2. For our trace, the version constant 0x0301 represents TLS 1.0

3. The Length covers only the payload of the Record Layer.

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.

Hello Messages

Find and inspect the details of the Client Hello and Server Hello messages, including expanding the Handshake 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.

Questions

Answer the following questions (answers are below).

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.

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.

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.

Answers

1. The random data is 28 bytes long for both client and server. (It does not include the timestamp, which is not random.)

2. The session ID sent by the server is 32 bytes long.

3. For our trace, the cipher method is TLS_RSA_WITH_RC4_128_SHA (0x0005).

Certificate Messages

Next, find and inspect the details of the Certificate message, including expanding the Handshake protocol 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:

1. *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.

Answer

1. The server sends a certificate to the client, since it is the browser that wants to verify the identity of the server. It is also possible for the server to request certificates from the client, but this behavior is not normally used by web applications.

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 (note, answers are on next page):

1. *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 Handshake Protocol.

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

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

Answers

1. The Client Key Exchange has a Content-Type of 22, indicating the Handshake protocol. This is the same as for the Hello and Certificate messages, as they are part of the Handshake protocol. The Change Cipher Spec message has a Content-Type of 20, indicating the Change Cipher Spec protocol. That is, this message is part of its own protocol and not the Handshake protocol.

2. Both sides send the Change Cipher Spec message immediately before they switch to sending encrypted contents. The message is an indication to the other side.

3. The contents of the Change Cipher Spec message are simply the value 1 as a single byte. Actually, it is the value "1" encrypted under the current scheme, which uses no encryption for the handshake so that we can see it.

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:

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

2. *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.

Answers to Alert Message

1. The Content-Type value is 21 for Alert. This is a new protocol, different from the Handshake, Change Cipher Spec and Application Data values that we have already seen.

2. The alert is encrypted; we cannot see its contents. Wireshark also describes the message as an "Encrypted Alert". Presumably is it a "close_notify" alert to signal that the connection is ending, but we cannot be certain.