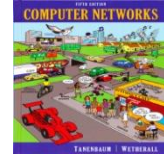


Lab Exercise – SSL/TLS



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.

The principal motivation for HTTPS is authentication of the accessed website and protection of the privacy and integrity of the exchanged data. It protects against man-in-the-middle attacks. The bidirectional encryption of communications between a client and server protects against eavesdropping and tampering of the communication. In practice, this provides a reasonable assurance that one is communicating without interference by attackers with the website that one intended to communicate with, as opposed to an impostor. Historically, HTTPS connections were primarily used for payment transactions on the World Wide Web, e-mail and for sensitive transactions in corporate information systems. Since 2018 HTTPS is more used on websites than the original non-secure HTTP; protecting page authenticity on all types of websites, securing accounts and keeping user communications, identity and web browsing private.

Step 1: Open a Trace

1. Open the Wireshark trace <https://kevincurran.org/com320/labs/wireshark/trace-ssl.pcap>

You should see the following trace.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.102	173.194.79.106	TCP	78	60245 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460
2	0.019644	173.194.79.106	192.168.1.102	TCP	74	443 → 60245 [SYN, ACK] Seq=0 Ack=1 Win=1460 Len=0
3	0.019829	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=1 Ack=1 Win=524288 Len=0
4	0.021328	192.168.1.102	173.194.79.106	TLSv1	186	Client Hello
5	0.040746	173.194.79.106	192.168.1.102	TCP	66	443 → 60245 [ACK] Seq=1 Ack=121 Win=14208 Len=0
6	0.041634	173.194.79.106	192.168.1.102	TLSv1	1484	Server Hello
7	0.041697	173.194.79.106	192.168.1.102	TLSv1	377	Certificate, Server Hello Done
8	0.041798	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=121 Ack=1730 Win=522928 Len=0
9	0.088543	192.168.1.102	173.194.79.106	TLSv1	252	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.105145	173.194.79.106	192.168.1.102	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message
11	0.105201	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=307 Ack=1777 Win=524288 Len=0
12	0.105436	192.168.1.102	173.194.79.106	TLSv1	239	Application Data
13	0.136468	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data
14	0.136525	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=480 Ack=3127 Win=523304 Len=0
15	0.137903	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data
16	0.137932	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=480 Ack=4477 Win=523304 Len=0
17	0.138469	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data, Application Data, Application Data
18	0.138500	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=480 Ack=5827 Win=523304 Len=0
19	0.138632	173.194.79.106	192.168.1.102	TLSv1	316	Application Data, Application Data
20	0.138660	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=480 Ack=6077 Win=524288 Len=0
21	0.140271	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data, Application Data
22	0.140309	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=480 Ack=7427 Win=523304 Len=0
23	0.144028	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data
24	0.144080	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=480 Ack=8777 Win=523304 Len=0
25	0.144465	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data
26	0.144490	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=480 Ack=10127 Win=523304 Len=0
27	0.150300	173.194.79.106	192.168.1.102	TLSv1	270	Application Data, Application Data
28	0.150461	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=480 Ack=10331 Win=524288 Len=0
29	0.150959	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data, Application Data
30	0.151093	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=480 Ack=11681 Win=523304 Len=0
31	0.155107	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data
32	0.155173	192.168.1.102	173.194.79.106	TCP	66	60245 → 443 [ACK] Seq=480 Ack=13031 Win=523304 Len=0

> Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface 0
> Ethernet II, Src: Apple_a2:05:1d (70:56:81:a2:05:1d), Dst: Cisco-Li_e3:e9:8d (00:16:b6:e3:e9:8d)
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 173.194.79.106
> Transmission Control Protocol, Src Port: 60245, Dst Port: 443, Seq: 0, Len: 0

Figure 1: Trace of “HTTPS” traffic

Step 2: Inspect the Trace

Now we are ready to look at the details of some “SSL” messages.

2. To begin, enter and apply a display filter of “ssl”. (see below)

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.

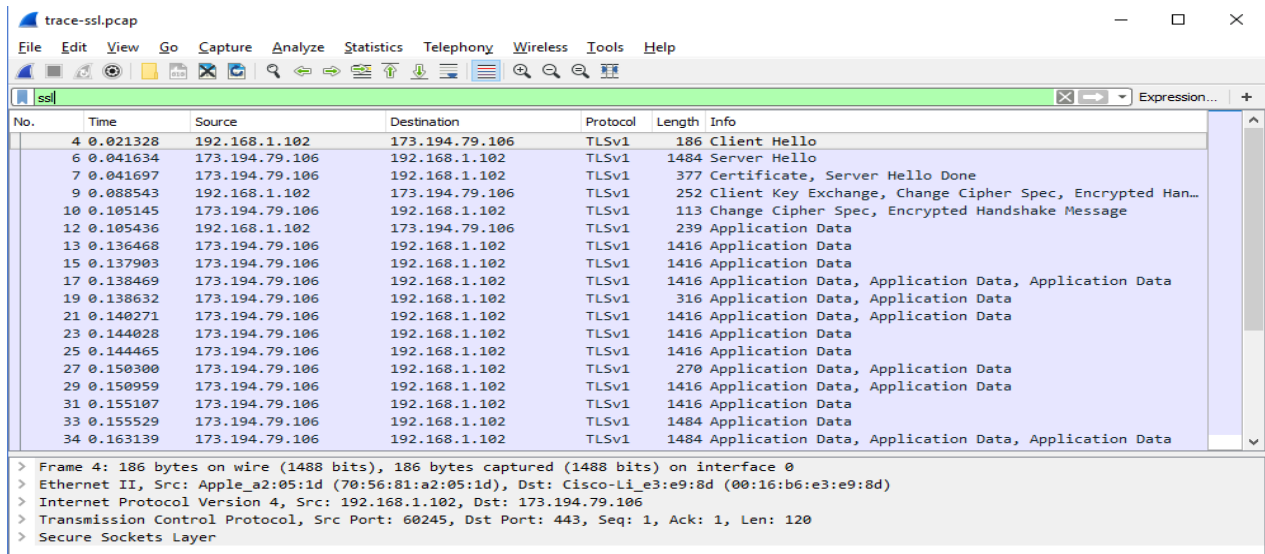
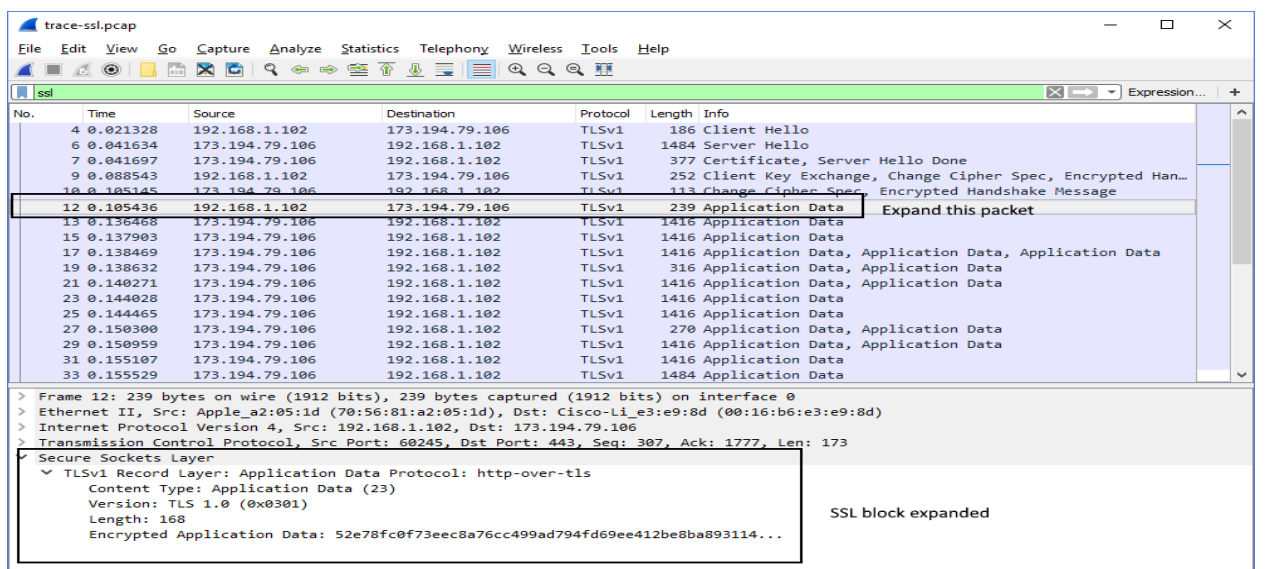


Figure 2: Trace of “SSL” traffic showing the details of the SSL header

3. Select a TLS message somewhere in the middle of your trace for which the Info reads “Applica-tion Data” & expand its Secure Sockets Layer block (by using the “+” expander or icon). For instance, packet #12 (see below).



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.

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.

The Content-Type for a record containing “Application Data” is 23. The version constant used in this trace is 0x0301 which represents TLS 1.0. 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:

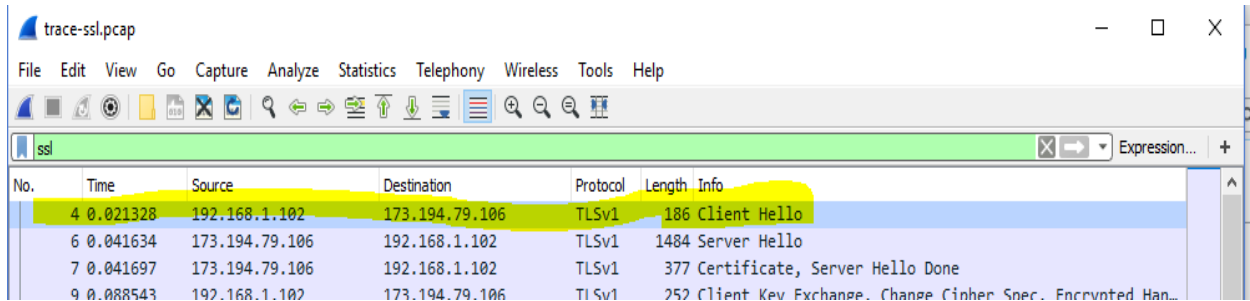
- a. Client (the browser) and Server (the web server) both send their Hellos
- b. Server sends its certificate to Client to authenticate (and optionally asks for Client Certificate)
- c. Client sends keying information and signals a switch to encrypted data.
- d. Server signals a switch to encrypted 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

Next we will 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.

4. Select packet #4, which is a TLS Client Hello message



No.	Time	Source	Destination	Protocol	Length	Info
4	0.021328	192.168.1.102	173.194.79.106	TLSv1	186	Client Hello
6	0.041634	173.194.79.106	192.168.1.102	TLSv1	1484	Server Hello
7	0.041697	173.194.79.106	192.168.1.102	TLSv1	377	Certificate, Server Hello Done
9	0.088543	192.168.1.102	173.194.79.106	TLSv1	252	Client Key Exchange, Change Cipher Spec, Encrypted Handshake

We can see several important fields here worth mentioning. First, the time (GMT seconds since midnight Jan 1, 1970) and random bytes (size 28) are included. This will be used later in the protocol to generate our symmetric encryption key. The client can send an optional session ID to quickly resume a previous TLS connection and skip portions of the TLS handshake. Arguably the most important part of the ClientHello message is the list of cipher suites, which dictate the key exchange algorithm, bulk encryption algorithm (with key length), MAC, and a pseudo-random function. The list should be ordered by client preference. The collection of these choices is a “cipher suite”, and the server is responsible for choosing a secure one it supports or return an error if it doesn’t support any. The final field specified in the specification is for compression methods. However, secure clients will advertise that they do not support compression (by passing “null” as the only algorithm) to avoid the CRIME attack. Finally, the ClientHello can have a number of different extensions. A common one is server_name, which specifies the host-name the connection is meant for, so web servers hosting multiple sites can present the correct certificate.

5. Select packet #6, which is a TLS Server Hello message

The session ID sent by the server is 32 bytes long. 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 (see below)

No.	Time	Source	Destination	Protocol	Length	Info
4	0.021328	192.168.1.102	173.194.79.106	TLSv1	186	Client Hello
6	0.041634	173.194.79.106	192.168.1.102	TLSv1	1484	Server Hello
7	0.041697	173.194.79.106	192.168.1.102	TLSv1	377	Certificate, Server Hello Done

Content Type: Handshake (22)

Version: TLS 1.0 (0x0301)

Length: 85

Handshake Protocol: Server Hello

Handshake Type: Server Hello (2)

Length: 81

Version: TLS 1.0 (0x0301)

Random: 501778d3d52d556ed20e072f638f0a51e9724d66ef5f1376...

GMT Unix Time: Jul 31, 2012 07:18:59.000000000 GMT Daylight Time

Random Bytes: d52d556ed20e072f638f0a51e9724d66ef5f13769d3a52e0...

Session ID Length: 32

Session ID: 8530bdac95116ccb343798b36cb2fd79c1e278cba1af4145...

The Cipher method chosen by the Server is *TLS_RSA_WITH_RC4_128_SHA (0x0005)*. The Client will list the different cipher methods it supports, and the Server will pick one of these methods to use.

```

Session ID Length: 32
Session ID: 8530bdac95116ccb343798b36cb2fd79c1e278cba1af4145...
Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
Compression Method: null (0)
Extensions Length: 9

```

Certificate Messages

- Next, find and inspect the details of the Certificate message including expanding the Handshake protocol block within the TLS Record (see below for expansion of packet #7).

No.	Time	Source	Destination	Protocol	Length	Info
4	0.021328	192.168.1.102	173.194.79.106	TLSv1	186	Client Hello
6	0.041634	173.194.79.106	192.168.1.102	TLSv1	1484	Server Hello
7	0.041697	173.194.79.106	192.168.1.102	TLSv1	377	Certificate, Server Hello Done
9	0.088543	192.168.1.102	173.194.79.106	TLSv1	252	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.105145	173.194.79.106	192.168.1.102	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message
12	0.105436	192.168.1.102	173.194.79.106	TLSv1	239	Application Data
13	0.136468	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data
15	0.137903	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data
17	0.138469	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data, Application Data, Application Data
19	0.138632	173.194.79.106	192.168.1.102	TLSv1	316	Application Data, Application Data
21	0.140271	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data, Application Data
23	0.144028	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data
25	0.144465	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data
27	0.150300	173.194.79.106	192.168.1.102	TLSv1	270	Application Data, Application Data
29	0.150959	173.194.79.106	192.168.1.102	TLSv1	1416	Application Data, Application Data

Secure Sockets Layer
TLSv1 Record Layer: Handshake Protocol: Certificate
Content Type: Handshake (22)
Version: TLS 1.0 (0x0301)
Length: 1625
Handshake Protocol: Certificate
Handshake Type: Certificate (11)
Length: 1621
Certificates Length: 1618
Certificates (1618 bytes)
Certificate Length: 805
Certificate: 308203213082028aa00302010202104f9d96d966b0992b54... (id-at-commonName=www.google.com,id-at-organizationName=Goo
Certificate Length: 807
Certificate: 308203233082028ca003020102020430000002300d06092a... (id-at-commonName=Thawte SGC CA,id-at-organizationName=Thaw

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.

Note it is the server that 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

7. Find and inspect the details of the Client Key Exchange and Change Cipher messages i.e. packet #9 (see below)

No.	Time	Source	Destination	Protocol	Length	Info
7	0.041697	173.194.79.106	192.168.1.102	TLSv1	377	Certificate, Server Hello Done
9	0.088543	192.168.1.102	173.194.79.106	TLSv1	252	Client Key Exchange, Change Cipher Spec, Encrypted Han...
10	0.105145	173.194.79.106	192.168.1.102	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message

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.

Note how 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.

```
✓ TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
  Content Type: Handshake (22)
  Version: TLS 1.0 (0x0301)
  Length: 134
  ✓ Handshake Protocol: Client Key Exchange
```

The Change Cipher Spec message has a Content-Type of 20, indicating the Change Cipher Spec protocol (see packet #10 – see below).

```
Secure Sockets Layer
  ✓ TLSv1 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
    Content Type: Change Cipher Spec (20)
    Version: TLS 1.0 (0x0301)
    Length: 1
    Change Cipher Spec Message
```

That is, this message is part of its own protocol and not the Handshake protocol.

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. 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

8. Finally, find and inspect the details of an Alert message at the end of the trace (packet #42).

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.

No.	Time	Source	Destination	Protocol	Length	Info
42	0.177209	192.168.1.102	173.194.79.106	TLSv1	93	Encrypted Alert

> Frame 42: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface 0
> Ethernet II, Src: Apple_a2:05:1d (70:56:81:a2:05:1d), Dst: Cisco-Li_e3:e9:8d (00:16:b6:e3:e9:8d)
> Internet Protocol Version 4, Src: 192.168.1.102, Dst: 173.194.79.106
> Transmission Control Protocol, Src Port: 60245, Dst Port: 443, Seq: 480, Ack: 20185, Len: 27
▼ Secure Sockets Layer
▼ TLSv1 Record Layer: Encrypted Alert
Content Type: Alert (21)
Version: TLS 1.0 (0x0301)
Length: 22
Alert Message: Encrypted Alert

Note, 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.

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.