

# TLS/SSL Handshake

## Analysis in Wireshark

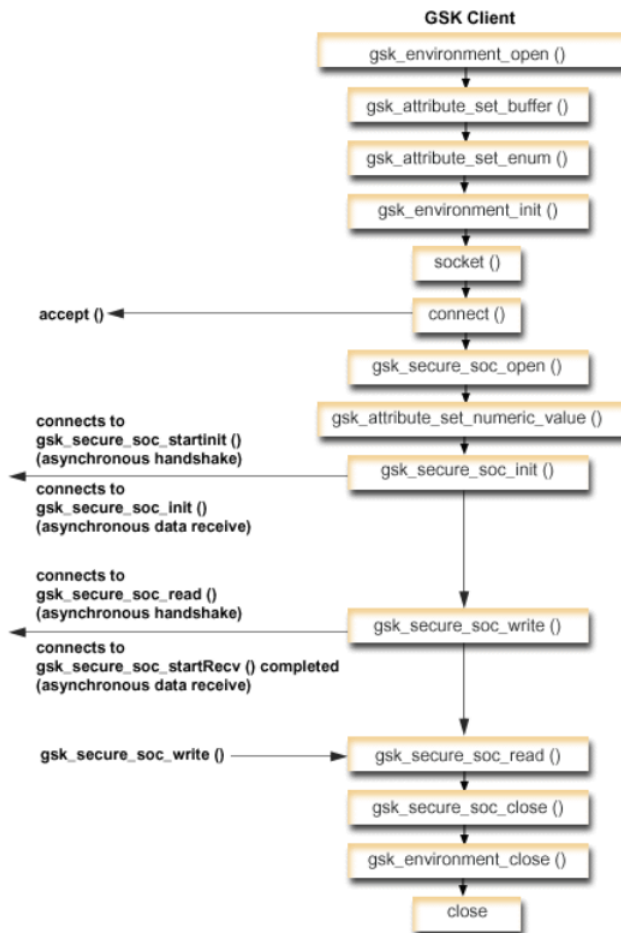
Lesley Nuttall

17 April 2019

Version 1

## SSL Connections

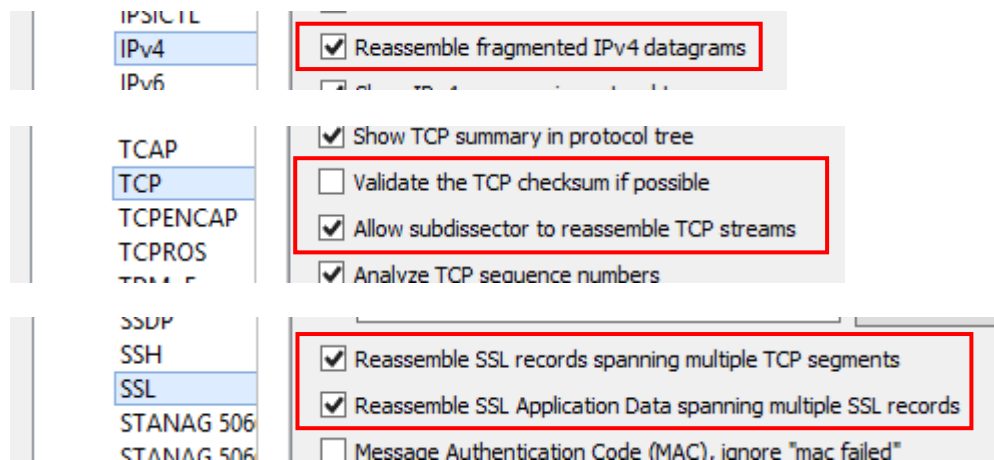
SSL/TLS is between the transport and the application layer and is protocol independent. Many IBM products make use of GSKit to establish a secure connection. The following graphic shows an example flow of API calls on a secure client using the GSKit APIs.



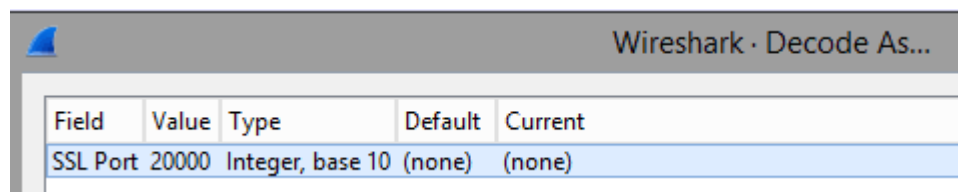
As you can see, the secure socket is opened after a `connect()` has been completed. In TCP conversations, you would see the SSL/TLS protocol traffic after a socket has been opened and a TCP connection successfully ESTABLISHED (as seen in netstat).

## Wireshark Setup

Edit -> Preferences -> Protocols



If none SSL/TLS ports are used and Wireshark does not recognise packets as SSL/TLS, you may need to define the port to be decoded as SSL. Can add this to 'Analyze -> Decode as':

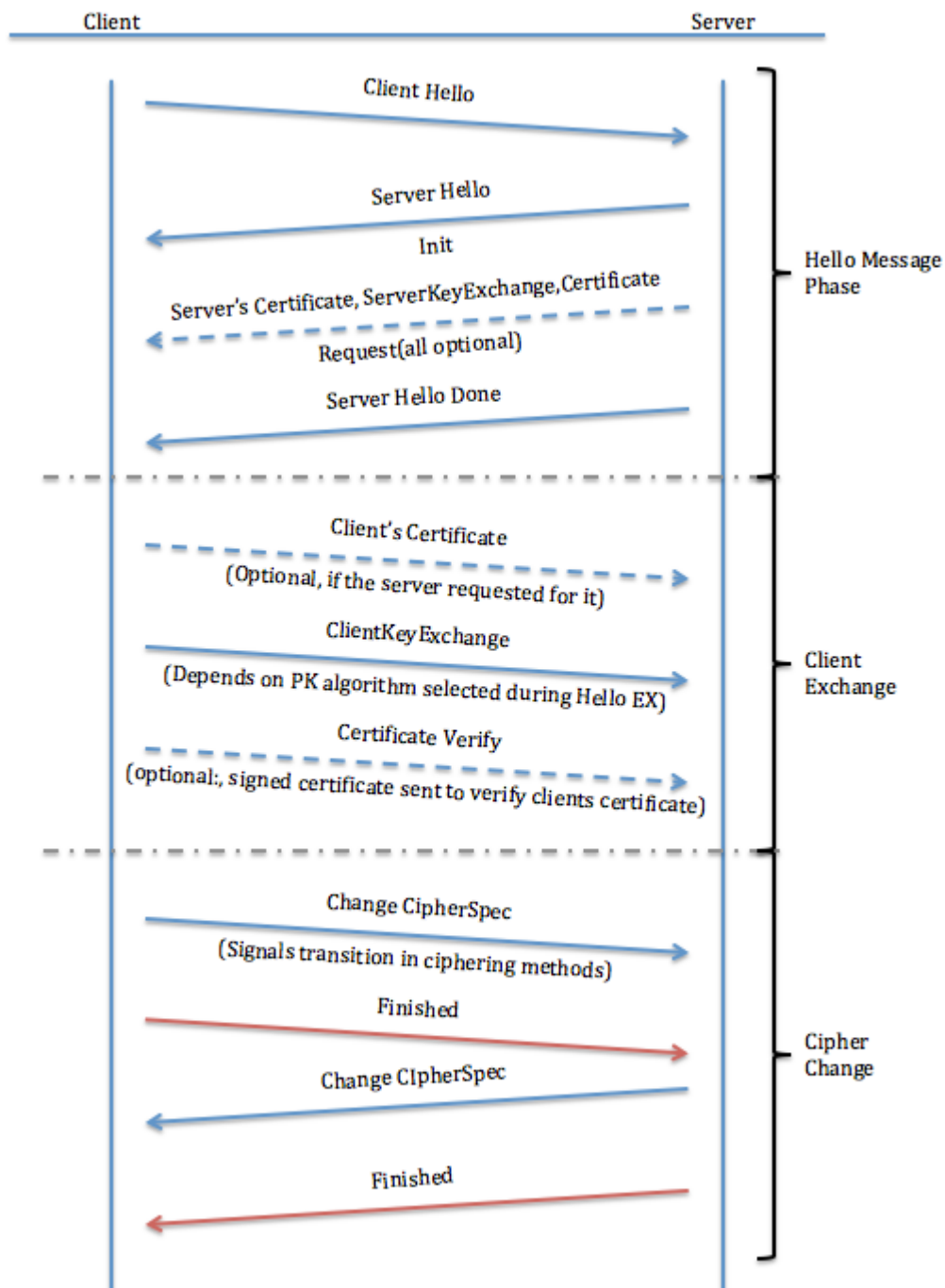


To keep track of SSL ID and SSL ID length, you can add two columns via Edit -> Preferences -> Columns, as follows:

Displayed	Title	Type	Fields
<input checked="" type="checkbox"/>	No.	Number	
<input checked="" type="checkbox"/>	Time	Time (format as specified)	
<input checked="" type="checkbox"/>	Delta	Delta time displayed	
<input checked="" type="checkbox"/>	Source	Source address	
<input checked="" type="checkbox"/>	Destination	Destination address	
<input checked="" type="checkbox"/>	Protocol	Protocol	
<input checked="" type="checkbox"/>	Length	Packet length (bytes)	
<input checked="" type="checkbox"/>	Info	Information	
<input checked="" type="checkbox"/>	SSL-ID	Custom	ssl.handshake.session_id
<input checked="" type="checkbox"/>	SSL-ID len	Custom	ssl.handshake.session_id_length

## SSL transaction flow

Source: ww.cisco.com



### The Hello Exchange

When an SSL client and server begin to communicate, they agree on a protocol version, select cryptographic algorithms, optionally authenticate each other, and use public key encryption techniques to generate shared secrets. These processes are performed in the handshake protocol. In summary, the client sends a Client Hello message to the server, which must respond with a Server Hello message or a fatal error occurs, and the connection fails.

## Interpreting a Client Hello

The Client Hello sends these attributes to the server:

- **Protocol Version:** The version of the SSL protocol by which the client wishes to communicate during this session.
- **Session ID:** The ID of a session the client wishes to use for this connection. In the first Client Hello of the exchange, the session ID is empty (as in the example below).
- **Cipher Suite:** The combinations of cryptographic algorithms supported by the client in order of the client's preference (first choice first). Each cipher suite defines both a key exchange algorithm and a cipher spec. The server selects a cipher suite or, if no acceptable choices are presented, returns a handshake failure alert and closes the connection.
- **Compression Method:** Includes a list of compression algorithms supported by the client. If the server does not support any method sent by the client, the connection fails. The compression method can also be null.

### Secure Sockets Layer

```
4 TLSv1 Record Layer: Handshake Protocol: Client Hello
  Content Type: Handshake (22)
  Version: TLS 1.0 (0x0301)
  Length: 133
4 Handshake Protocol: Client Hello
  Handshake Type: Client Hello (1)
  Length: 129
  Version: TLS 1.0 (0x0301)
  Random: 5c38954b6d7ac9a26408f3c2c72e2cdc8d5fe77d01034993...
  Session ID Length: 0
  Cipher Suites Length: 28
4 Cipher Suites (14 suites)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
  Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
  Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
  Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
  Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
  Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
  Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
  Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
  Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x0032)
  Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
  Cipher Suite: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x0013)
  Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
  Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
  Compression Methods Length: 1
4 Compression Methods (1 method)
  Compression Method: null (0)
  Extensions Length: 60
  Extension: server_name (len=22)
  Extension: status_request (len=5)
  Extension: supported_groups (len=6)
  Extension: ec_point_formats (len=2)
  Extension: extended_master_secret (len=0)
  Extension: renegotiation_info (len=1)
```

## Interpreting a Server Hello

You can view information about the certificates presented during an SSL handshake. In the example Server Hello below, you can see the following:

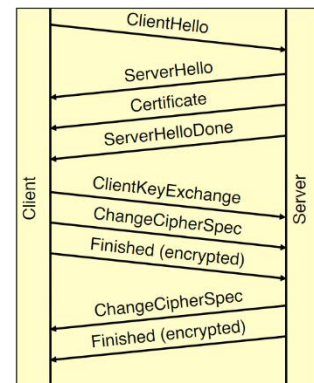
- **Protocol Version:** The chosen version of the SSL protocol that the client supports.
- **Session ID:** This is the identity of the session that corresponds to this connection. If the session ID sent by the client in the Client Hello is not empty, the server looks in the session cache for a match. If a match is found and the server is willing to establish the new connection using the specified session state, the server responds with the same value that was supplied by the client. This indicates a resumed session and dictates that the parties must proceed directly to the finished messages. Otherwise, this field contains a different value that identifies the new session. The server might return an empty **session\_id** to indicate that the session will not be cached, and therefore cannot be resumed.
- **Cipher Suite:** As selected by the server from the list that was sent from the client.
- **Compression Method:** As selected by the server from the list that was sent from the client.
- **Certificate Request:** The server sends the client a list of all the certificates that are configured on it and allows the client to select which certificate it wants to use for authentication.

```
Secure Sockets Layer
4 TLSv1.2 Record Layer: Handshake Protocol: Multiple Handshake Messages
  Content Type: Handshake (22)
  Version: TLS 1.2 (0x0303)
  Length: 4625
4 Handshake Protocol: Server Hello
  Handshake Type: Server Hello (2)
  Length: 77
  Version: TLS 1.2 (0x0303)
  Random: 2f59006123e2886b4880506a3b1aee976301f8ec3e5639b7...
  Session ID Length: 32
  Session ID: 2a2137e1506822846ed0f4f8445c99a212c5b3dbe46277b6...
  Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
  Compression Method: null (0)
  Extensions Length: 5
  Extension: renegotiation_info (len=1)
4 Handshake Protocol: Certificate
  Handshake Type: Certificate (11)
  Length: 4536
  Certificates Length: 4533
4 Certificates (4533 bytes)
  Certificate Length: 1482
  Certificate: 308205c6308203aea003020102021336000030143777f99... (id-at-commonName=
  Certificate Length: 1643
  Certificate: 308206673082044fa00302010202137b00000027b04a471... (id-at-commonName=
  Certificate Length: 1399
  Certificate: 308205733082035ba003020102021047b3bbe82718c28f42... (id-at-commonName=
4 Handshake Protocol: Server Hello Done
  Handshake Type: Server Hello Done (14)
  Length: 0
```

## New SSL connection

This is a fresh connection from a client and it will not have any reference to any previous SSL session id, so will normally set the 'Session ID length' field to 0. The server will establish a new session, by performing a full SSL handshake and negotiating keys with the client. The server will assign an SSL session ID that the client will remember. The server application would also normally have an SSL session cache that it maintains, with a list of all SSL session ids.

Protocol	Length	Info
TCP	74	22454 → 4000 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2617
TCP	74	4000 → 22454 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1
TCP	66	22454 → 4000 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2617138731 TSecr=16
TLSv1.2	258	Client Hello
TCP	66	4000 → 22454 [ACK] Seq=1 Ack=193 Win=30080 Len=0 TSval=160563734 TSecr=2
TLSv1.2	4704	Server Hello, Certificate, Server Hello Done
TCP	66	22454 → 4000 [ACK] Seq=193 Ack=4639 Win=38528 Len=0 TSval=2617138895 TSecr=2
TLSv1.2	384	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
TCP	66	4000 → 22454 [ACK] Seq=4639 Ack=511 Win=31104 Len=0 TSval=160563903 TSecr=2
TLSv1.2	72	Change Cipher Spec
TLSv1.2	111	Encrypted Handshake Message



In the first Client Hello of an exchange, the session ID is empty. You can search for these packets using the display filter "ssl.handshake.type == 1 and ssl.handshake.session\_id\_length == 0"

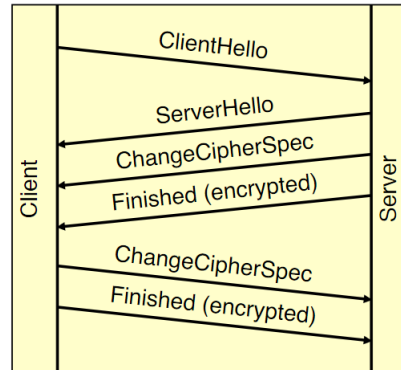
No.	Time	Delta	Source	Destination	Protocol	Length	Info
49752	7.205934	0.000666			TLSv1.2	170	Client Hello
49831	7.238793	0.032859			TLSv1.2	170	Client Hello
50129	7.284068	0.045275			TLSv1.2	170	Client Hello
50250	7.305478	0.021410			TLSv1.2	170	Client Hello
50319	7.311263	0.005785			TLSv1.2	170	Client Hello
51857	7.418130	0.106867			TLSv1	184	Client Hello
52287	7.452057	0.033927			TLSv1.2	170	Client Hello
53481	7.564377	0.112320			TLSv1.2	170	Client Hello
53612	7.575901	0.011524			TLSv1.2	170	Client Hello
53712	7.584647	0.008746			TLSv1.2	170	Client Hello
54277	7.636559	0.051912			TLSv1.2	170	Client Hello
54979	7.735826	0.099267			TLSv1.2	170	Client Hello

- 4 TLSv1.2 Record Layer: Handshake Protocol: Client Hello
  - Content Type: Handshake (22)
  - Version: TLS 1.2 (0x0303)
  - Length: 99
  - 4 Handshake Protocol: Client Hello
    - Handshake Type: Client Hello (1)
    - Length: 95
    - Version: TLS 1.2 (0x0303)
    - Random: 4dbbe3d4c242d0a086906f5760864e8c088860fd990d3b48...
    - Session ID Length: 0
    - Cipher Suites Length: 36
    - ▶ Cipher Suites (18 suites)
    - Compression Methods Length: 1
    - ▶ Compression Methods (1 method)
    - Extensions Length: 18
    - ▶ Extension: supported\_groups (len=8)
    - ▶ Extension: ec\_point\_formats (len=2)

## Resumed SSL connection

If the Client Hello presents a session id that the Server recognises, the session is resumed – meaning the session reuses previously negotiated keys. A full SSL handshake is not required. This is quicker and less ‘expensive’ than full key negotiation.

Protocol	Length	Info
TCP	74	21012 → 7135 [SYN] Seq=0 Win=29
TCP	74	7135 → 21012 [SYN, ACK] Seq=0 A
TCP	66	21012 → 7135 [ACK] Seq=1 Ack=1
TLSv1.2	200	Client Hello
TCP	66	7135 → 21012 [ACK] Seq=1 Ack=13
TLSv1.2	158	Server Hello
TCP	66	21012 → 7135 [ACK] Seq=135 Ack=
TLSv1.2	72	Change Cipher Spec
TCP	66	21012 → 7135 [ACK] Seq=135 Ack=
TLSv1.2	72	Change Cipher Spec
TLSv1.2	111	Encrypted Handshake Message



You can search for these packets using the display filter “ssl.resume”. Wireshark expert info also tells you the session reuses previously negotiated keys.

### 4 Secure Sockets Layer

- 4 TLSv1.2 Record Layer: Handshake Protocol: Client Hello
  - Content Type: Handshake (22)
  - Version: TLS 1.2 (0x0303)
  - Length: 219
  - 4 Handshake Protocol: Client Hello
    - Handshake Type: Client Hello (1)
    - Length: 215
    - Version: TLS 1.2 (0x0303)
    - Random: 5c38954d85177492431f3ee9fdcedccef8ade9457ca9cb0...
    - Session ID Length: 32
    - Session ID: b6afcefa33613eed905ba7a739d75ba5b301639527190eb0...
    - Cipher Suites Length: 42
    - Cipher Suites (21 suites)
    - Compression Methods Length: 1
    - Compression Methods (1 method)

### 4 Secure Sockets Layer

- 4 TLSv1.2 Record Layer: Handshake Protocol: Server Hello
  - Content Type: Handshake (22)
  - Version: TLS 1.2 (0x0303)
  - Length: 85
  - 4 Handshake Protocol: Server Hello
    - Handshake Type: Server Hello (2)
    - Length: 81
    - Version: TLS 1.2 (0x0303)
    - Random: d9b36e5ad82a37a68dcc8dcef7e7b841144b7abb9987c77d...
    - Session ID Length: 32
    - Session ID: b6afcefa33613eed905ba7a739d75ba5b301639527190eb0...
    - Cipher Suite: TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256 (0x009c)
    - Compression Method: null (0)
    - Extensions Length: 9
    - Extension: extended\_master\_secret (len=0)

### 4 Secure Sockets Layer

- 4 TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  - Content Type: Change Cipher Spec (20)
  - Version: TLS 1.2 (0x0303)
  - Length: 1
  - 4 Change Cipher Spec Message
    - Expert Info (Note/Sequence): This session reuses previously negotiated keys (Session resumption)
    - [This session reuses previously negotiated keys (Session resumption)]
    - [Severity level: Note]
    - [Group: Sequence]



## Renegotiated SSL connection

The client thinks it is re-using an SSL session, so it will present the previously used SSL session ID to the server. If the server does not recognise the session ID, a full SSL handshake is required in order to negotiate keys again. As a new session must be negotiated, the Server Hello response will contain a new SSL ID that does not match the one presented in the Client Hello. This is resource intensive (similar to a new ssl session).

The server may not recognise the SSL id for various reasons - the server ssl cache is full, the server ssl cache timeout has triggered, the ssl session id has been invalidated in the cache, in a cluster/load balanced env the connection has switched to a new server, etc.

Protocol	Length	Info	SSL-ID
TCP	74	38544 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=160557762 ...	
TCP	74	443 → 38544 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1...	
TCP	66	38544 → 443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=160557763 TSecr=150780522	
TLSv1.2	202	Client Hello	3ce258265124bc406d5d1cb2d...
TCP	1514	443 → 38544 [ACK] Seq=1 Ack=137 Win=131584 Len=1448 TSval=150780574 TSecr=160...	
TCP	66	38544 → 443 [ACK] Seq=137 Ack=1449 Win=32128 Len=0 TSval=160558282 TSecr=1507...	
TCP	1514	443 → 38544 [ACK] Seq=1449 Ack=137 Win=131584 Len=1448 TSval=150780574 TSecr=...	
TCP	66	38544 → 443 [ACK] Seq=137 Ack=2897 Win=35072 Len=0 TSval=160558282 TSecr=1507...	
TCP	1514	443 → 38544 [ACK] Seq=2897 Ack=137 Win=131584 Len=1448 TSval=150780574 TSecr=...	
TCP	66	38544 → 443 [ACK] Seq=137 Ack=4345 Win=37888 Len=0 TSval=160558282 TSecr=1507...	
TLSv1.2	353	Server Hello, Certificate, Server Hello Done	eea15d7f8140336929af06ab9...
TCP	66	38544 → 443 [ACK] Seq=137 Ack=4632 Win=40832 Len=0 TSval=160558282 TSecr=1507...	
TLSv1.2	333	Client Key Exchange	
TLSv1.2	72	Change Cipher Spec	
TCP	66	443 → 38544 [ACK] Seq=4632 Ack=410 Win=131328 Len=0 TSval=150780691 TSecr=160...	
TLSv1.2	111	Encrypted Handshake Message	
TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message	

### 4 Handshake Protocol: Client Hello

Handshake Type: Client Hello (1)

Length: 127

Version: TLS 1.2 (0x0303)

Random: 51e7b018d4efeb65e2ebbc2afb6bd1ab5238d0758468fc30...

Session ID Length: 32

Session ID: 3ce258265124bc406d5d1cb2dd7d1ecff29c686d8afd8c27...

Cipher Suites Length: 36

Cipher Suites (18 suites)

### 4 Handshake Protocol: Server Hello

Handshake Type: Server Hello (2)

Length: 77

Version: TLS 1.2 (0x0303)

Random: fc4d44c46d99abe8bf047c2061c636f1d2810efbbee98d48...

Session ID Length: 32

Session ID: eea15d7f8140336929af06ab9d2ba3ff6c9144606f97ef2b...

Cipher Suite: TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256 (0x009c)

## Example delay from establishing TCP connection to SSL handshake

0.82 second delay from establishing connection to sending ssl handshake

No.	Time	Delta	Source	Destination	Protocol	Length	Info
78243	10.522361	0.000000			TCP	74	38054 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=160562042 TSecr=0 WS=128
82442	11.345329	0.822968			TLSv1.1	202	Client Hello
82515	11.348342	0.003013			TLSv1.1	227	Server Hello, Change Cipher Spec, Encrypted Handshake Message
82516	11.348355	0.000013			TCP	66	38054 → 443 [ACK] Seq=137 Ack=162 Win=30336 Len=0 TSval=160562868 TSecr=150663138
82602	11.362448	0.014093			TLSv1.1	72	Change Cipher Spec
84436	11.537555	0.175107			TLSv1.1	135	Encrypted Handshake Message
84466	11.540515	0.002960			TCP	66	443 → 38054 [ACK] Seq=162 Ack=212 Win=514 Len=0 TSval=150663157 TSecr=160562882

0.45 second delay from establishing connection to sending ssl handshake

No.	Time	Delta	Source	Destination	Protocol	Length	Info
62779	13:08:30.375521	0.000000			TCP	74	37426 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=160560190 TSecr=0 WS=128
64487	13:08:30.541385	0.165864			TCP	74	443 → 37426 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=150662877 TSecr=16056...
64488	13:08:30.541403	0.000018			TCP	66	37426 → 443 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=160560356 TSecr=150662877
66327	13:08:30.986943	0.445540			TLSv1.1	202	Client Hello
66337	13:08:30.993620	0.006677			TLSv1.1	227	Server Hello, Change Cipher Spec, Encrypted Handshake Message
66338	13:08:30.993634	0.000014			TCP	66	37426 → 443 [ACK] Seq=137 Ack=162 Win=30336 Len=0 TSval=160560808 TSecr=150662932
66375	13:08:31.012265	0.010631			TLSv1.1	72	Change Cipher Spec
67376	13:08:31.214608	0.202343			TCP	66	443 → 37426 [ACK] Seq=162 Ack=143 Win=131584 Len=0 TSval=150662954 TSecr=160560827
67523	13:08:31.233115	0.018507			TLSv1.1	135	Encrypted Handshake Message
67951	13:08:31.276130	0.043015			TLSv1.1	1063	Application Data
67963	13:08:31.276956	0.000826			TCP	66	443 → 37426 [ACK] Seq=162 Ack=1209 Win=130560 Len=0 TSval=150662960 TSecr=160561847
68245	13:08:31.303681	0.026725			TLSv1.1	919	Application Data
68667	13:08:31.343163	0.039482			TCP	66	37426 → 443 [ACK] Seq=1209 Ack=1015 Win=32000 Len=0 TSval=160561158 TSecr=150662963
68784	13:08:31.353488	0.010325			TCP	66	443 → 37426 [FIN, ACK] Seq=1015 Ack=1209 Win=130560 Len=0 TSval=150662967 TSecr=160561158
71141	13:08:31.570564	0.212076			TCP	66	37426 → 443 [FIN, ACK] Seq=1209 Ack=1016 Win=32000 Len=0 TSval=160561385 TSecr=150662967
71460	13:08:31.597830	0.027266			TCP	66	443 → 37426 [ACK] Seq=1016 Ack=1210 Win=130560 Len=0 TSval=150662992 TSecr=160561385

This could indicate that there is a performance issue on the server acting as the SSL client. Key negotiation is resource expensive, so check to see if sessions are not being resumed where you would expect them to be (i.e. there are many unnecessary renegotiated SSL handshakes).