SSL

“When we visit a secure website like Amazon (which uses HTTPS), our browser and the website create a secure connection using the SSL/TLS protocol.

In Wireshark, we can capture this process — it’s called the **SSL/TLS handshake**.
You’ll see packets like **‘Client Hello’** and **‘Server Hello’**.

- The **Client Hello** message is sent by your computer to say, ‘Hello, I want to connect securely, and here are the encryption methods I support.’
- The **Server Hello** comes from the website (server) saying, ‘Hello, I accept your request and will use this encryption method.”

After that, they exchange keys to create a **secure, encrypted communication channel**, so no one else can read the data. This is how websites keep our passwords and banking information safe.”

for example if you get command like (**tcp.stream eq 2)**

tcp.stream eq 2

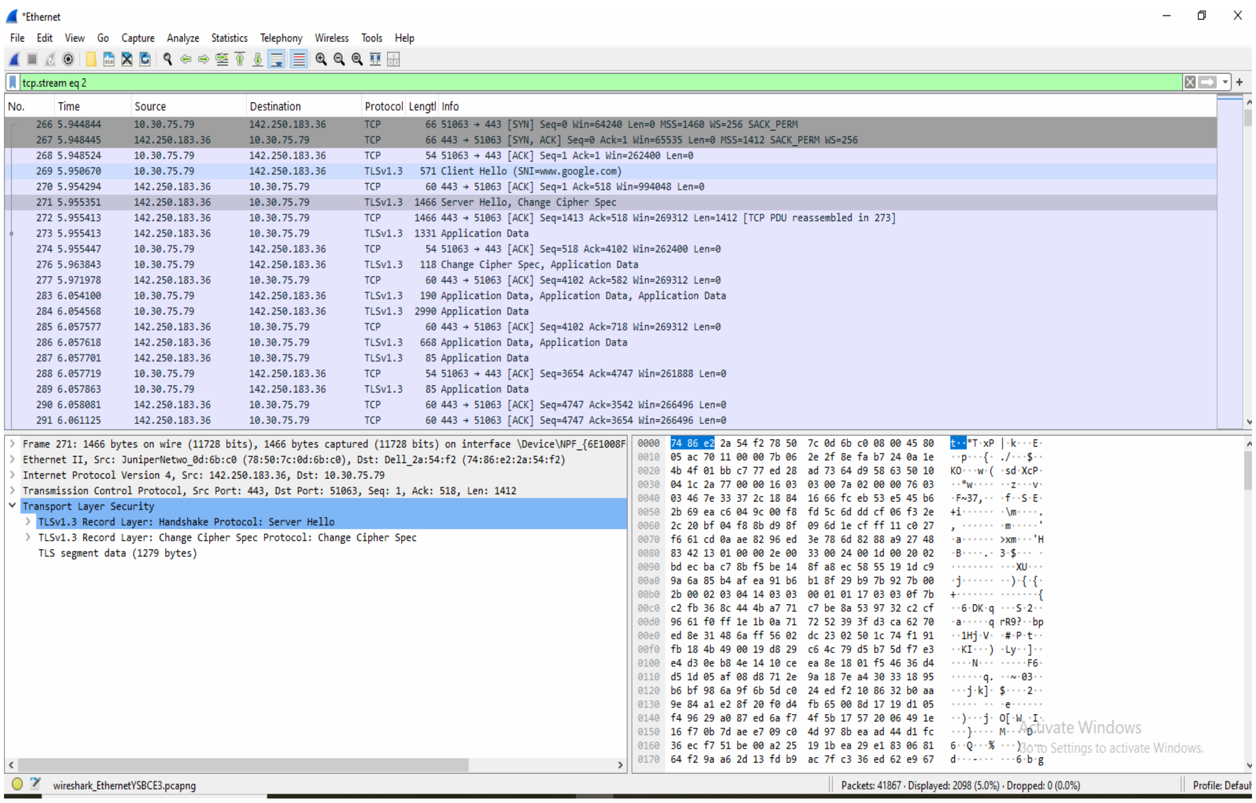
This is a **Wireshark display filter**.
It means:

“Show only the packets that belong to **TCP connection number 2**.”

Each TCP connection (or “stream”) between your computer and a server gets a number (0, 1, 2, etc.).
So when you type tcp.stream eq 2, Wireshark only shows packets from that conversation — between your system (client) and the website (server, e.g. Amazon).

It helps you **focus on one secure session** instead of seeing all network traffic.

While Running On My Machine I found following details which may will get change as per machine varies


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp.stream eq 2

No.  Time      Source          Destination     Protocol Lengtl Info
 266 5.944844  10.30.75.79     142.250.183.36  TCP        66 51063 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
 267 5.948445  142.250.183.36  10.30.75.79     TCP        66 443 → 51063 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1412 SACK_PERM WS=256
 268 5.948524  10.30.75.79     142.250.183.36  TCP        54 51063 → 443 [ACK] Seq=1 Ack=1 Win=262400 Len=0
 269 5.950670  10.30.75.79     142.250.183.36  TLSv1.3   571 Client Hello (SNI=www.google.com)
 270 5.954294  142.250.183.36  10.30.75.79     TCP        60 443 → 51063 [ACK] Seq=1 Ack=518 Win=994048 Len=0
 271 5.955351  142.250.183.36  10.30.75.79     TLSv1.3  1466 Server Hello, Change Cipher Spec
 272 5.955413  142.250.183.36  10.30.75.79     TCP      1466 443 → 51063 [ACK] Seq=1413 Ack=518 Win=269312 Len=1412 [TCP PDU reassembled in 273]
 273 5.955413  142.250.183.36  10.30.75.79     TLSv1.3  1331 Application Data
 274 5.955447  10.30.75.79     142.250.183.36  TCP        54 51063 → 443 [ACK] Seq=518 Ack=4102 Win=262400 Len=0
 276 5.963843  10.30.75.79     142.250.183.36  TLSv1.3   118 Change Cipher Spec, Application Data
 277 5.971978  142.250.183.36  10.30.75.79     TCP        60 443 → 51063 [ACK] Seq=4102 Ack=582 Win=269312 Len=0
 283 6.054100  10.30.75.79     142.250.183.36  TLSv1.3   190 Application Data, Application Data, Application Data
 284 6.054568  10.30.75.79     142.250.183.36  TLSv1.3  2990 Application Data
 285 6.057577  142.250.183.36  10.30.75.79     TCP        60 443 → 51063 [ACK] Seq=4102 Ack=718 Win=269312 Len=0
 286 6.057618  142.250.183.36  10.30.75.79     TLSv1.3   668 Application Data, Application Data
 287 6.057701  142.250.183.36  10.30.75.79     TLSv1.3    85 Application Data
 288 6.057719  10.30.75.79     142.250.183.36  TCP        54 51063 → 443 [ACK] Seq=3654 Ack=4747 Win=261888 Len=0
 289 6.057863  10.30.75.79     142.250.183.36  TLSv1.3    85 Application Data
 290 6.058081  142.250.183.36  10.30.75.79     TCP        60 443 → 51063 [ACK] Seq=4747 Ack=3542 Win=266496 Len=0
 291 6.061125  142.250.183.36  10.30.75.79     TCP        60 443 → 51063 [ACK] Seq=4747 Ack=3654 Win=266496 Len=0
```

```
> Frame 271: 1466 bytes on wire (11728 bits), 1466 bytes captured (11728 bits) on interface \Device\NPF_{6E1008F
> Ethernet II, Src: JuniperNetwo_0d:6b:c0 (78:50:7c:0d:6b:c0), Dst: Dell_2a:54:f2 (74:86:e2:2a:54:f2)
> Internet Protocol Version 4, Src: 142.250.183.36, Dst: 10.30.75.79
> Transmission Control Protocol, Src Port: 443, Dst Port: 51063, Seq: 1, Ack: 518, Len: 1412
v Transport Layer Security
  > TLSv1.3 Record Layer: Handshake Protocol: Server Hello
  > TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
    TLS segment data (1279 bytes)
```

```
0000  74 86 e2 2a 54 f2 78 50  7c 0d 6b c0 08 00 45 80   t··*T·xP |·k···E·
0010  05 ac 70 11 00 00 7b 06  2e 2f 8e fa b7 24 0a 1e   ··p···{· ./···$··
0020  4b 4f 01 bb c7 77 ed 28  ad 73 64 d9 58 63 50 10   KO···w·( ·sd·XcP·
0030  04 1c 2a 77 00 00 16 03  03 00 7a 02 00 00 76 03   ··*w···· ··z···v·
0040  03 46 7e 33 37 2c 18 84  16 66 fc eb 53 e5 45 b6   ·F~37,·· ·f··S·E·
0050  2b 69 ea c6 04 9c 00 f8  fd 5c 6d dd cf 06 f3 2e   +i······ ·\m····.
0060  2c 20 bf 04 f8 8b d9 8f  09 6d 1e cf ff 11 c0 27   , ······ ·m·····'
0070  f6 61 cd 0a ae 82 96 ed  3e 78 6d 82 88 a9 27 48   ·a······ >xm···'H
0080  83 42 13 01 00 00 2e 00  33 00 24 00 1d 00 20 02   ·B····.· 3·$··· ·
0090  bd ec ba c7 8b f5 be 14  8f a8 ec 58 55 19 1d c9   ········ ···XU···
00a0  9a 6a 85 b4 af ea 91 b6  b1 8f 29 b9 7b 92 7b 00   ·j······ ··)·{·{·
00b0  2b 00 02 03 04 14 03 03  00 01 01 17 03 03 0f 7b   +······· ·······{
00c0  c2 fb 36 8c 44 4b a7 71  c7 be 8a 53 97 32 c2 cf   ··6·DK·q ···S·2··
00d0  96 61 f0 ff 1e 1b 0a 71  72 52 39 3f d3 ca 62 70   ·a·····q rR9?··bp
00e0  ed 8e 31 48 6a ff 56 02  dc 23 02 50 1c 74 f1 91   ··1Hj·V· ·#·P·t··
00f0  fb 18 4b 49 00 19 d8 29  c6 4c 79 d5 b7 5d f7 e3   ··KI···) ·Ly··]··
0100  e4 d3 0e b8 4e 14 10 ce  ea 8e 18 01 f5 46 36 d4   ····N··· ·····F6·
0110  d5 1d 05 af 08 d8 71 2e  9a 18 7e a4 30 33 18 95   ······q. ··~·03··
0120  b6 bf 98 6a 9f 6b 5d c0  24 ed f2 10 86 32 b0 aa   ···j·k]· $····2··
0130  9e 84 a1 e2 8f 20 f0 d4  fb 65 00 8d 17 19 d1 05   ········ ·e······
0140  f4 96 29 a0 87 ed 6a f7  4f 5b 17 57 20 06 49 1e   ··)···j· O[·W ·I·
0150  16 f7 0b 7d ae e7 09 c0  4d 97 8b ea ad 44 d1 fc   ···}···· M····D··
0160  36 ec f7 51 be 00 a2 25  19 1b ea 29 e1 83 06 81   6··Q···% ···)····
0170  64 f2 9a a6 2d 13 fd b9  ac 7f c3 36 ed 62 e9 67   d···-··· ···6·b·g
```

wireshark_EthernetYSBCE3.pcapng
Packets: 41867 · Displayed: 2098 (5.0%) · Dropped: 0 (0.0%)
Profile: Defaul
Activate Windows
Go to Settings to activate Windows.

the fields you shown in Wireshark window you must know

1. Frame : This line tells us about the **whole packet** Wireshark captured.

   Example:
   Frame 271: 1466 bytes on wire
   means this packet is 1466 bytes in total size.

2. Ethernet II: This is the **Data Link Layer (Layer 2)** of the OSI model.
   It shows information like the **MAC addresses** of sender and receiver devices.
   Example:
   Src: → the source device (your computer)
   Dst: → the destination (router or server)
   So, **Ethernet II = physical network address info**.

3. Internet Protocol Version 4 (IPv4)

This is the **Network Layer (Layer 3)**.It shows the **IP addresses** of the source and destination.

Example:

a. `Src: 142.250.183.36` → the server (like Amazon or Google)
b. `Dst: 10.30.75.79` → your computer's IP

So, **IPv4 = shows who is talking to whom over the internet**.

4. Transmission Control Protocol (TCP) :

This is the **Transport Layer (Layer 4)**.It handles reliable delivery of data.

Example:

- Src Port: 443 → server port for HTTPS
- Dst Port: 51063 → your computer's port

TCP = ensures safe and ordered data delivery.

5. Transport Layer Security (TLS)

TLSv1.3 Record Layer: Handshake Protocol: Server Hello

- This is part of the **SSL/TLS handshake**.
- It means the **server is replying** to the client's "Hello".
- The server tells:
  - "Yes, I accept secure connection."
  - "Here's the encryption method I'll use."

So, **Server Hello = Server's response to create a secure connection.**

6. **TLSv1.3 Record Layer: Change Cipher Spec Protocol**

This means both sides are **switching from plain text to encrypted communication**.
After this message, all data becomes **encrypted** (cannot be read in Wireshark).
So, **Change Cipher Spec = Start of encryption**
In short you must keep in your mind
Frame → Whole captured packet

Ethernet II → Device (MAC) addresses

IPv4 → IP addresses of sender and receiver

TCP → Transport control (port numbers, sequence)

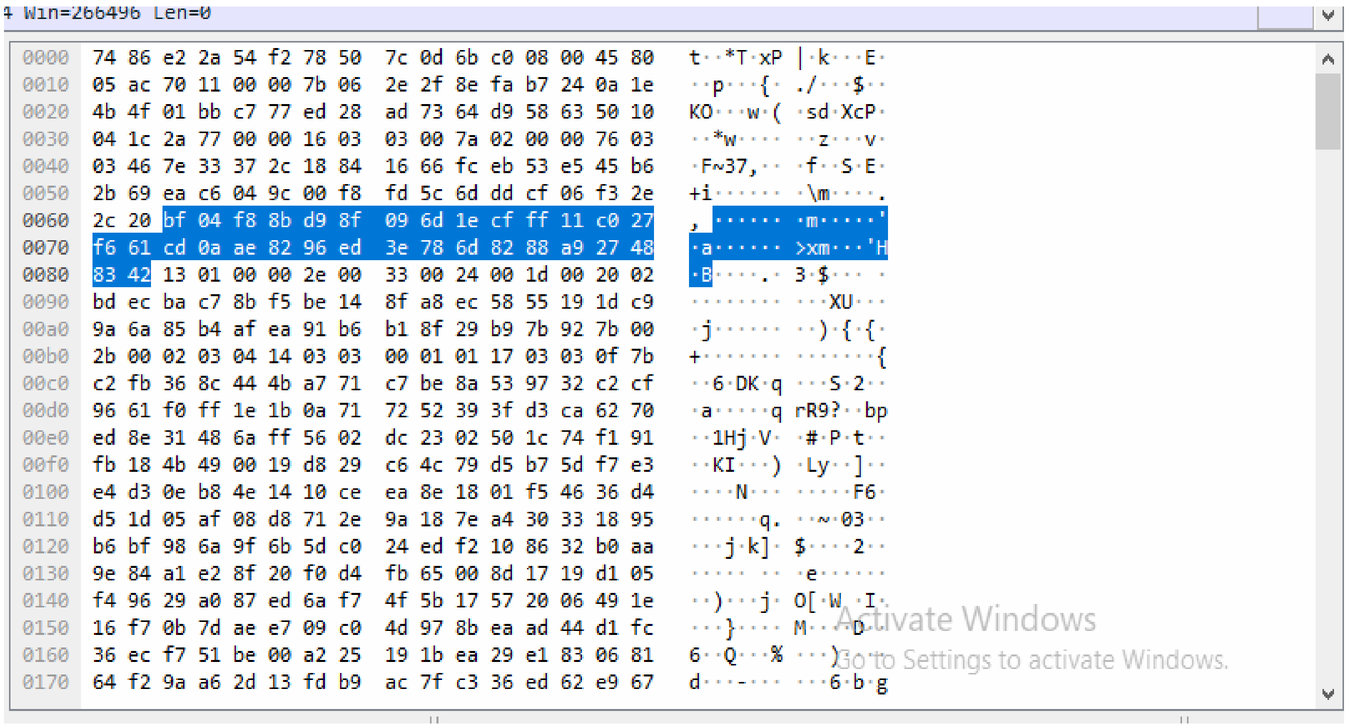
TLS → Security layer (handshake and encryption)

same explanation applies to the “Client Hello” packet — just the roles are reversed.

TLSv1.3 Record Layer: Handshake Protocol: Client Hello

This packet is sent **from your computer (the client)** to the **server (like Amazon or Google)** when you first open a secure website.

How to Read This



```
4 Win=266496 Len=0

0000  74 86 e2 2a 54 f2 78 50  7c 0d 6b c0 08 00 45 80   t··*T·xP |·k···E·
0010  05 ac 70 11 00 00 7b 06  2e 2f 8e fa b7 24 0a 1e   ··p···{· ./···$··
0020  4b 4f 01 bb c7 77 ed 28  ad 73 64 d9 58 63 50 10   KO···w·( ·sd·XcP·
0030  04 1c 2a 77 00 00 16 03  03 00 7a 02 00 00 76 03   ··*w···· ··z···v·
0040  03 46 7e 33 37 2c 18 84  16 66 fc eb 53 e5 45 b6   ·F~37,·· ·f··S·E·
0050  2b 69 ea c6 04 9c 00 f8  fd 5c 6d dd cf 06 f3 2e   +i······ ·\m····.
0060  2c 20 bf 04 f8 8b d9 8f  09 6d 1e cf ff 11 c0 27   , ······ ·m·····'
0070  f6 61 cd 0a ae 82 96 ed  3e 78 6d 82 88 a9 27 48   ·a······ >xm···'H
0080  83 42 13 01 00 00 2e 00  33 00 24 00 1d 00 20 02   ·B····.· 3·$··· ·
0090  bd ec ba c7 8b f5 be 14  8f a8 ec 58 55 19 1d c9   ········ ···XU···
00a0  9a 6a 85 b4 af ea 91 b6  b1 8f 29 b9 7b 92 7b 00   ·j······ ··)·{·{·
00b0  2b 00 02 03 04 14 03 03  00 01 01 17 03 03 0f 7b   +······· ·······{
00c0  c2 fb 36 8c 44 4b a7 71  c7 be 8a 53 97 32 c2 cf   ··6·DK·q ···S·2··
00d0  96 61 f0 ff 1e 1b 0a 71  72 52 39 3f d3 ca 62 70   ·a·····q rR9?··bp
00e0  ed 8e 31 48 6a ff 56 02  dc 23 02 50 1c 74 f1 91   ··1Hj·V· ·#·P·t··
00f0  fb 18 4b 49 00 19 d8 29  c6 4c 79 d5 b7 5d f7 e3   ··KI···) ·Ly··]··
0100  e4 d3 0e b8 4e 14 10 ce  ea 8e 18 01 f5 46 36 d4   ····N··· ·····F6·
0110  d5 1d 05 af 08 d8 71 2e  9a 18 7e a4 30 33 18 95   ······q. ··~·03··
0120  b6 bf 98 6a 9f 6b 5d c0  24 ed f2 10 86 32 b0 aa   ···j·k]· $····2··
0130  9e 84 a1 e2 8f 20 f0 d4  fb 65 00 8d 17 19 d1 05   ····· ·· ·e······
0140  f4 96 29 a0 87 ed 6a f7  4f 5b 17 57 20 06 49 1e   ··)···j· O[·W ·I·
0150  16 f7 0b 7d ae e7 09 c0  4d 97 8b ea ad 44 d1 fc   ···}···· M····D··
0160  36 ec f7 51 be 00 a2 25  19 1b ea 29 e1 83 06 81   6··Q···% ···)····
0170  64 f2 9a a6 2d 13 fd b9  ac 7f c3 36 ed 62 e9 67   d···-··· ···6·b·g
```

Part	**Description**	**Example from your image**

Left side	Memory offset (line number in hex).	0000, 0010, 0020 ...
Middle part	Actual data in **hexadecimal**. Each pair (like 74 86) = 1 byte.	74 86 e2 e2 54 f2 78 50 ...
Right side	ASCII (text) view of that data, if printable.	You see things like t..T.xP..

This is the raw packet data. Wireshark shows it in hexadecimal form.
Each pair of numbers represents one byte of data.
After encryption, this data looks like random letters and numbers — this is how SSL keeps our information safe.”