Date:28.11.2022

Final Year B. Tech., Sem VII 2022-23

Cryptography And Network Security Lab

Assignment submission

PRN No: 2019BTECS00036

Full name: Nikhil Danapgol

Batch: B2

Assignment: 17

Title of assignment: SSL/TLS Handshake Analysis using Wireshark

Title:

SSL/TLS Handshake Analysis using Wireshark

Aim:

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

Theory:

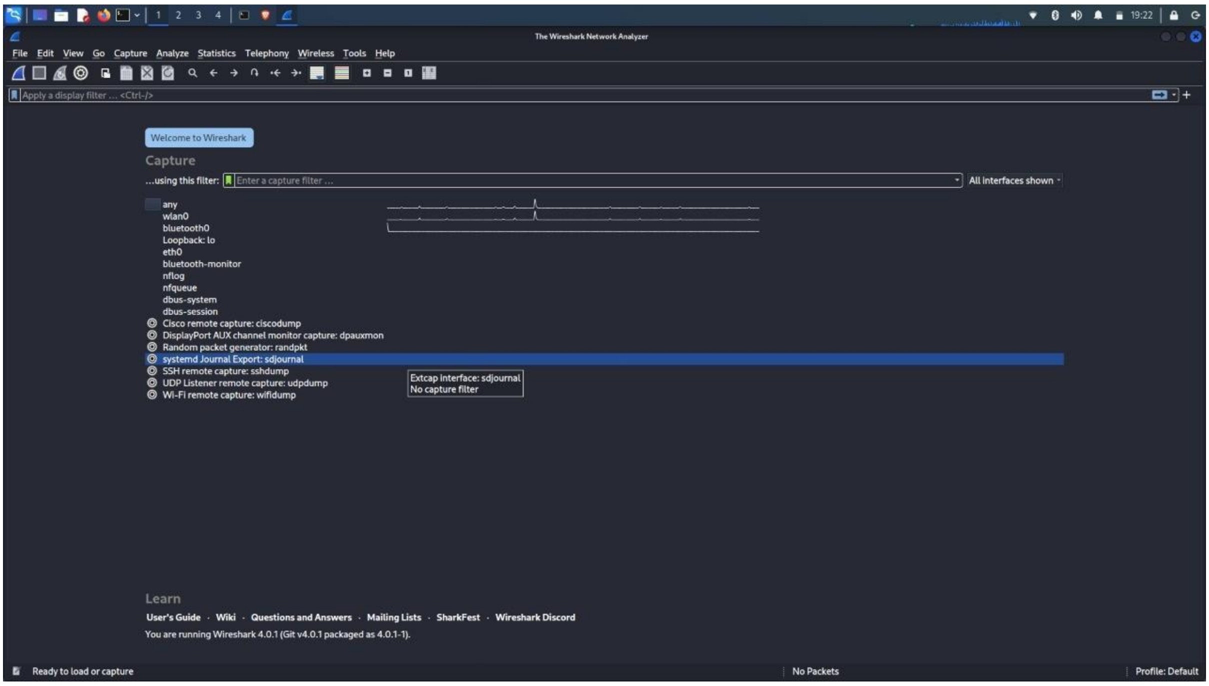
- Wireshark is a free and open-source packet analyzer.
- It is used for network troubleshooting, analysis, software and communications protocol development, and education.
- Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues.

- Wireshark is cross-platform, using the Qt widget toolkit in current releases to implement its user interface, and using pcap to capture packets; it runs on Linux, macOS, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows.
- There is also a terminal-based (non-GUI) version called TShark. Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of the GNU General Public License version 2 or any later version.

Use of Wireshark

Step 1: Open a Trace you should use a supplied trace file trace-ssl.pcap.

File → Open → open from folder containing file

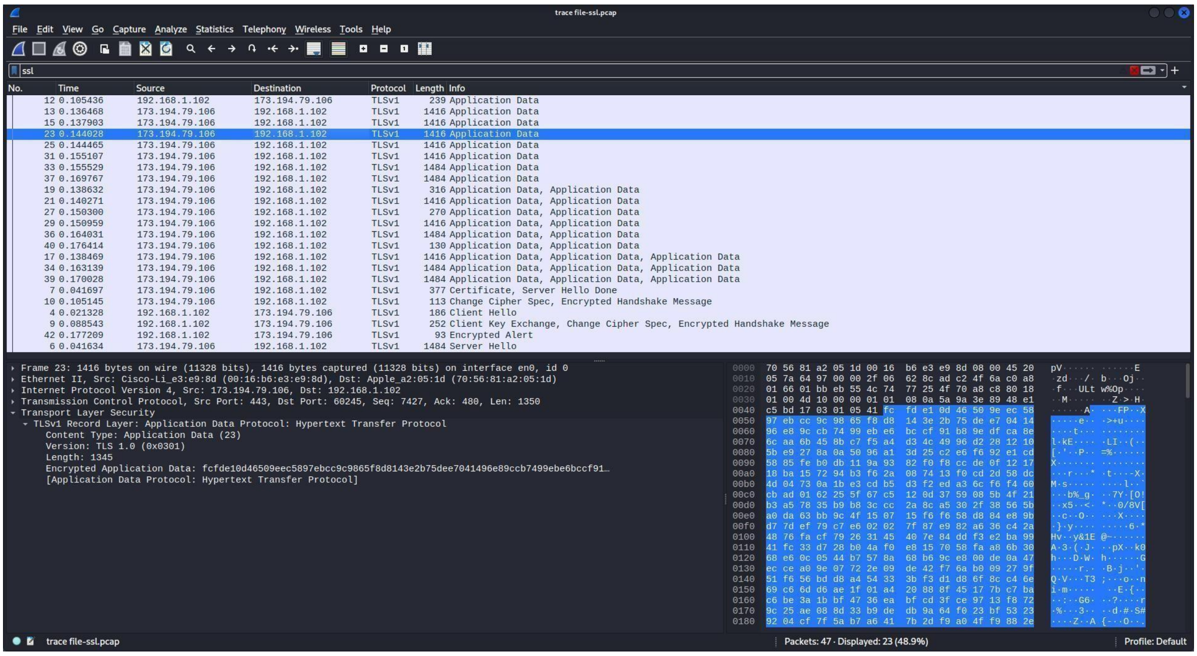

1 2 3 4
19:22
The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Welcome to Wireshark
Capture
…using this filter:
Enter a capture filter …
All interfaces shown
any
wlan0
bluetooth0
Loopback: lo
eth0
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt
systemd Journal Export: sdjournal
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
Wi-Fi remote capture: wifidump
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord

Extcap interface: sdjournal
No capture filter

You are running Wireshark 4.0.1 (Git v4.0.1 packaged as 4.0.1-1).

Ready to load or capture | No Packets | Profile: Default

Applying SSL Filter


trace file-ssl.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ssl

```
No.   Time      Source          Destination     Protocol Length Info
   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
   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
   31 0.155107  173.194.79.106  192.168.1.102   TLSv1    1416 Application Data
   33 0.155529  173.194.79.106  192.168.1.102   TLSv1    1484 Application Data
   37 0.169767  173.194.79.106  192.168.1.102   TLSv1    1484 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
   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
   36 0.164031  173.194.79.106  192.168.1.102   TLSv1    1484 Application Data, Application Data
   40 0.176414  173.194.79.106  192.168.1.102   TLSv1     130 Application Data, Application Data
   17 0.138469  173.194.79.106  192.168.1.102   TLSv1    1416 Application Data, Application Data, Application Data
   34 0.163139  173.194.79.106  192.168.1.102   TLSv1    1484 Application Data, Application Data, Application Data
   39 0.170028  173.194.79.106  192.168.1.102   TLSv1    1484 Application Data, Application Data, Application Data
    7 0.041697  173.194.79.106  192.168.1.102   TLSv1     377 Certificate, Server Hello Done
   10 0.105145  173.194.79.106  192.168.1.102   TLSv1     113 Change Cipher Spec, Encrypted Handshake Message
    4 0.021328  192.168.1.102   173.194.79.106  TLSv1     186 Client Hello
    9 0.088543  192.168.1.102   173.194.79.106  TLSv1     252 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
   42 0.177209  192.168.1.102   173.194.79.106  TLSv1      93 Encrypted Alert
    6 0.041634  173.194.79.106  192.168.1.102   TLSv1    1484 Server Hello
```

```
› Frame 23: 1416 bytes on wire (11328 bits), 1416 bytes captured (11328 bits) on interface en0, id 0
› Ethernet II, Src: Cisco-Li_e3:e9:8d (00:16:b6:e3:e9:8d), Dst: Apple_a2:05:1d (70:56:81:a2:05:1d)
› Internet Protocol Version 4, Src: 173.194.79.106, Dst: 192.168.1.102
› Transmission Control Protocol, Src Port: 443, Dst Port: 60245, Seq: 7427, Ack: 480, Len: 1350
⌄ Transport Layer Security
  ⌄ TLSv1 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 1345
      Encrypted Application Data: fcfde10d46509eec5897ebcc9c9865f8d8143e2b75dee7041496e89ccb7499ebe6bccf91…
      [Application Data Protocol: Hypertext Transfer Protocol]
```

```
0000  70 56 81 a2 05 1d 00 16  b6 e3 e9 8d 08 00 45 20   pV······ ······E 
0010  05 7a 64 97 00 00 2f 06  62 8c ad c2 4f 6a c0 a8   ·zd···/· b···Oj··
0020  01 66 01 bb eb 55 4c 74  77 25 4f 70 a8 c8 80 18   ·f···ULt w%Op····
0030  01 00 4d 10 00 00 01 01  08 0a 5a 9a 3e 89 48 e1   ··M····· ··Z·>·H·
0040  c5 bd 17 03 01 05 41 fc  fd e1 0d 46 50 9e ec 58   ······A· ···FP··X
0050  97 eb cc 9c 98 65 f8 d8  14 3e 2b 75 de e7 04 14   ·····e·· ·>+u····
0060  96 e8 9c cb 74 99 eb e6  bc cf 91 b8 9e df ca 8e   ····t··· ········
0070  6c aa 6b 45 8b c7 f5 a4  d3 4c 49 96 d2 28 12 10   l·kE···· ·LI··(··
0080  5b e9 27 8a 0a 50 96 a1  3d 25 c2 e6 f6 92 e1 cd   [·'··P·· =%······
0090  58 85 fe b0 db 11 9a 93  82 f0 f8 cc de 0f 12 17   X······· ········
00a0  18 ba 15 72 94 b3 f6 2a  08 74 13 f0 cd 2d 58 dc   ···r···* ·t···-X·
00b0  4d 04 73 0a 1b e3 cd b5  d3 f2 ed a3 6c f6 f4 60   M·s····· ····l··`
00c0  cb ad 01 62 25 5f 67 c5  12 0d 37 59 08 5b 4f 21   ···b%_g· ··7Y·[O!
00d0  b3 a5 78 35 b9 b8 3c cc  2a 8c a5 30 2f 38 56 5b   ··x5··<· *··0/8V[
00e0  a0 da 63 bb 9c 4f 15 07  15 f6 f6 58 d8 84 e8 9b   ··c··O·· ···X····
00f0  d7 7d ef 79 c7 e6 02 02  7f 87 e9 82 a6 36 c4 2a   ·}·y···· ·····6·*
0100  48 76 fa cf 79 26 31 45  40 7e 84 dd f3 e2 ba 99   Hv··y&1E @~······
0110  41 fc 33 d7 28 b0 4a f0  e8 15 70 58 fa a8 6b 30   A·3·(·J· ··pX··k0
0120  68 e6 0c 05 44 b7 57 8a  68 b6 9c e8 00 de 0a 47   h···D·W· h······G
0130  ec ce a0 9e 07 72 2e 09  de 42 f7 6a b0 09 27 9f   ·····r.· ·B·j··'·
0140  51 f6 56 bd d8 a4 54 33  3b f3 d1 d8 6f 8c c4 6e   Q·V···T3 ;···o··n
0150  69 c6 6d d6 ae 1f 01 a4  20 88 8f 45 17 7b c7 ba   i·m·····  ··E·{··
0160  c6 be 3a 1b bf 47 36 ea  bf cd 3f ce 97 13 f8 72   ··:··G6· ··?····r
0170  9c 25 ae 08 8d 33 b9 de  db 9a 64 f0 23 bf 53 23   ·%···3·· ··d·#·S#
0180  92 04 cf 7f 5a b7 a6 41  7b 2d f9 a0 4f f9 88 2e   ····Z··A {-··O··.
```

Packets: 47 · Displayed: 23 (48.9%)   Profile: Default

Step 2: Inspect the Trace

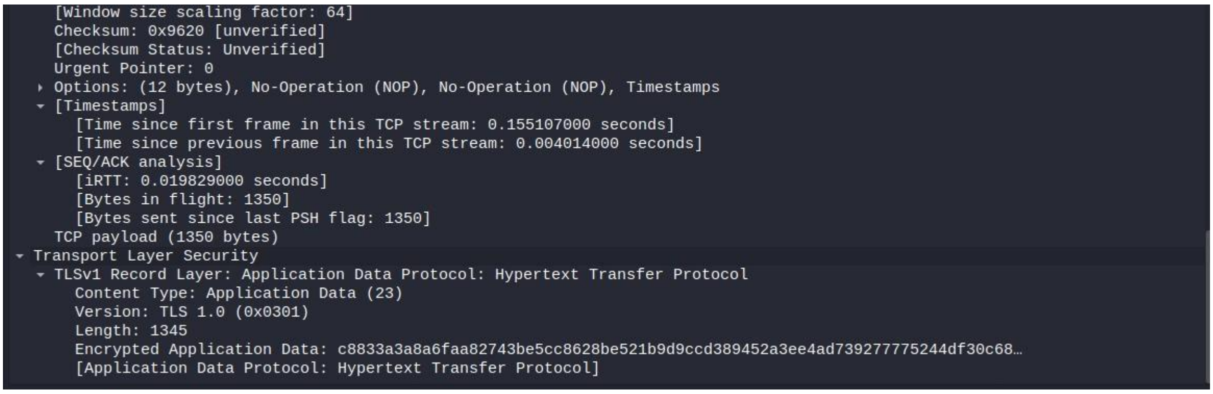
Now we are ready to look at the details of some SSL messages. To begin, enter and apply a display filter of ssl. This filter will help to simplify the display by showing only SSL and TLS messages. It will exclude other TCP segments that are part of the trace, such as Acks and connection open/close. Select a TLS message somewhere in the middle of your trace for which the Info field reads Application Data, and expand its Secure Sockets Layer block(by using triangular icon on left side). 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.

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.

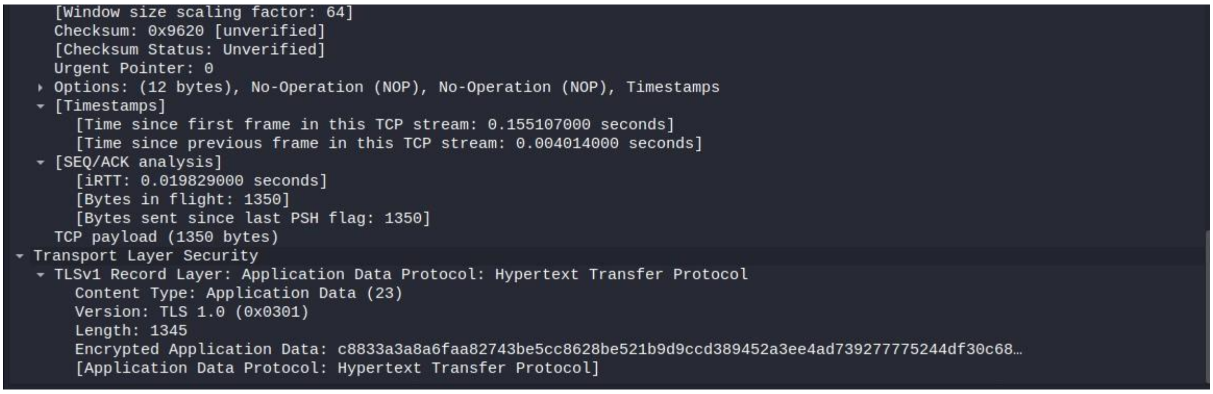
1. What is the Content Type for a record containing Application Data? Ans:Aplication Data



```
    [Window size scaling factor: 64]
    Checksum: 0x9620 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▾ [Timestamps]
      [Time since first frame in this TCP stream: 0.155107000 seconds]
      [Time since previous frame in this TCP stream: 0.004014000 seconds]
  ▾ [SEQ/ACK analysis]
      [iRTT: 0.019829000 seconds]
      [Bytes in flight: 1350]
      [Bytes sent since last PSH flag: 1350]
    TCP payload (1350 bytes)
▾ Transport Layer Security
  ▾ TLSv1 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 1345
      Encrypted Application Data: c8833a3a8a6faa82743be5cc8628be521b9d9ccd389452a3ee4ad739277775244df30c68…
      [Application Data Protocol: Hypertext Transfer Protocol]
```

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

Ans:1.0



```
    [Window size scaling factor: 64]
    Checksum: 0x9620 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▾ [Timestamps]
      [Time since first frame in this TCP stream: 0.155107000 seconds]
      [Time since previous frame in this TCP stream: 0.004014000 seconds]
  ▾ [SEQ/ACK analysis]
      [iRTT: 0.019829000 seconds]
      [Bytes in flight: 1350]
      [Bytes sent since last PSH flag: 1350]
    TCP payload (1350 bytes)
▾ Transport Layer Security
  ▾ TLSv1 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
      Content Type: Application Data (23)
      Version: TLS 1.0 (0x0301)
      Length: 1345
      Encrypted Application Data: c8833a3a8a6faa82743be5cc8628be521b9d9ccd389452a3ee4ad739277775244df30c68…
      [Application Data Protocol: Hypertext Transfer Protocol]
```

Step 3: 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:

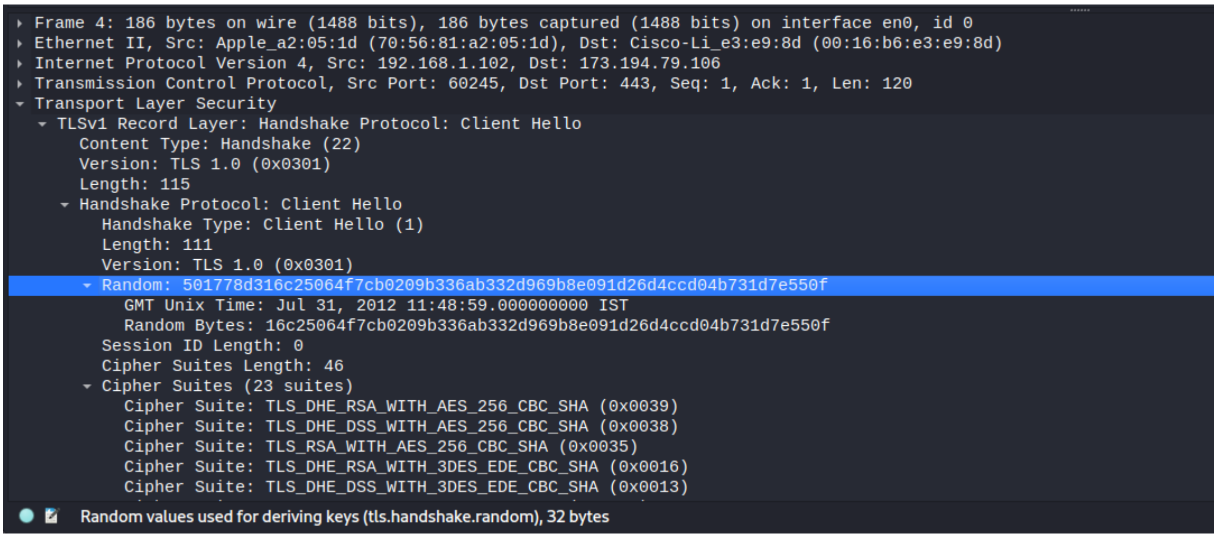
- Client (the browser) and Server(the web server) both send their Hellos
- Server sends its certificate to Client to authenticate (and optionally asks for Client Certificate)
- Client sends keying information and signals a switch to encrypted data.
- Server signals a switch to encrypted data.
- Both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

Hello Message

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.

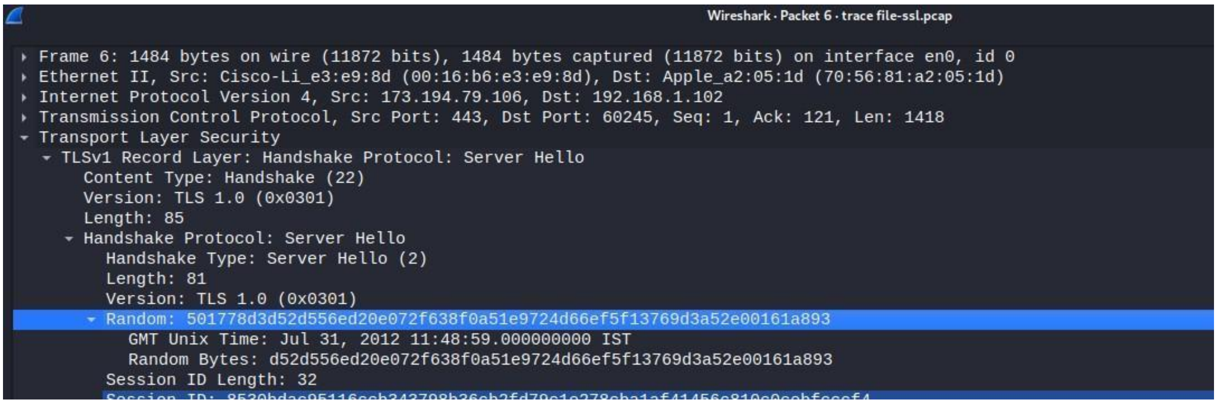
1. How long in bytes is the random data in the Hellos? Both the Client and Server include this random data (a nonce) to allow the establishment of session keys.

Ans:



```
▸ Frame 4: 186 bytes on wire (1488 bits), 186 bytes captured (1488 bits) on interface en0, id 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: 1, Ack: 1, Len: 120
▾ Transport Layer Security
  ▾ TLSv1 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 115
    ▾ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 111
        Version: TLS 1.0 (0x0301)
      ▾ Random: 501778d316c25064f7cb0209b336ab332d969b8e091d26d4ccd04b731d7e550f
          GMT Unix Time: Jul 31, 2012 11:48:59.000000000 IST
          Random Bytes: 16c25064f7cb0209b336ab332d969b8e091d26d4ccd04b731d7e550f
        Session ID Length: 0
        Cipher Suites Length: 46
      ▾ Cipher Suites (23 suites)
          Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
          Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
          Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
          Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x0016)
          Cipher Suite: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x0013)
```

Random values used for deriving keys (tls.handshake.random), 32 bytes

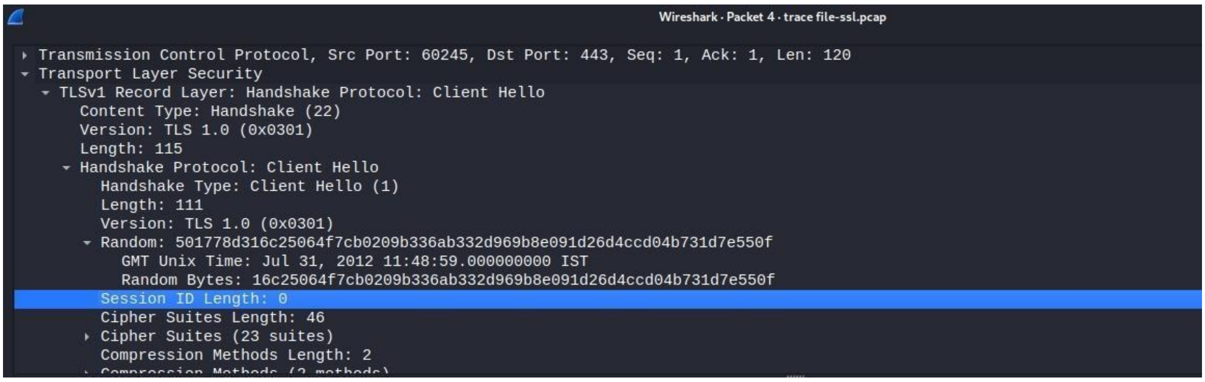


Wireshark · Packet 6 · trace file-ssl.pcap

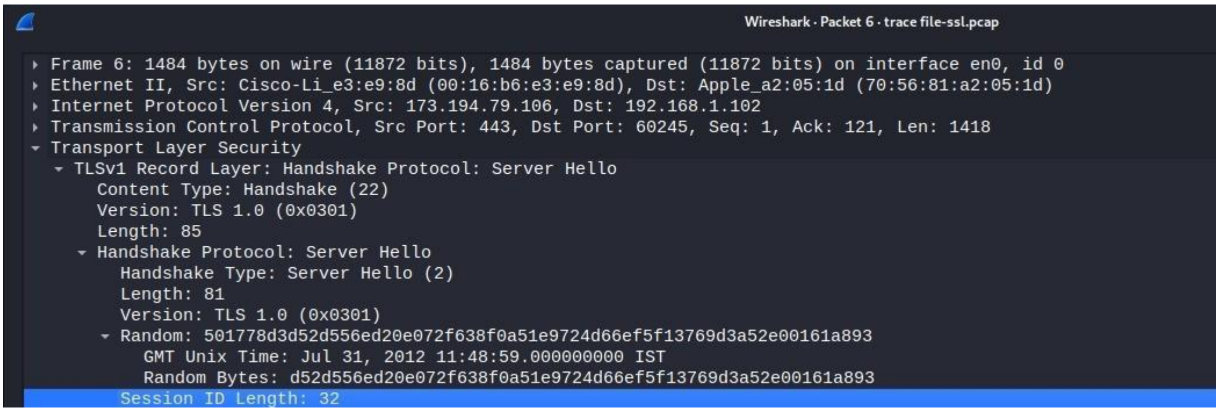
```
▸ Frame 6: 1484 bytes on wire (11872 bits), 1484 bytes captured (11872 bits) on interface en0, id 0
▸ Ethernet II, Src: Cisco-Li_e3:e9:8d (00:16:b6:e3:e9:8d), Dst: Apple_a2:05:1d (70:56:81:a2:05:1d)
▸ Internet Protocol Version 4, Src: 173.194.79.106, Dst: 192.168.1.102
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 60245, Seq: 1, Ack: 121, Len: 1418
▾ Transport Layer Security
  ▾ TLSv1 Record Layer: Handshake Protocol: Server Hello
      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: 501778d3d52d556ed20e072f638f0a51e9724d66ef5f13769d3a52e00161a893
          GMT Unix Time: Jul 31, 2012 11:48:59.000000000 IST
          Random Bytes: d52d556ed20e072f638f0a51e9724d66ef5f13769d3a52e00161a893
        Session ID Length: 32
        Session ID: 8530bdac95116ecb343798b36cb2fd79e1e278cba1af41456c810c0cebfcccf4
```

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.

Ans:0(client) & 32(server)



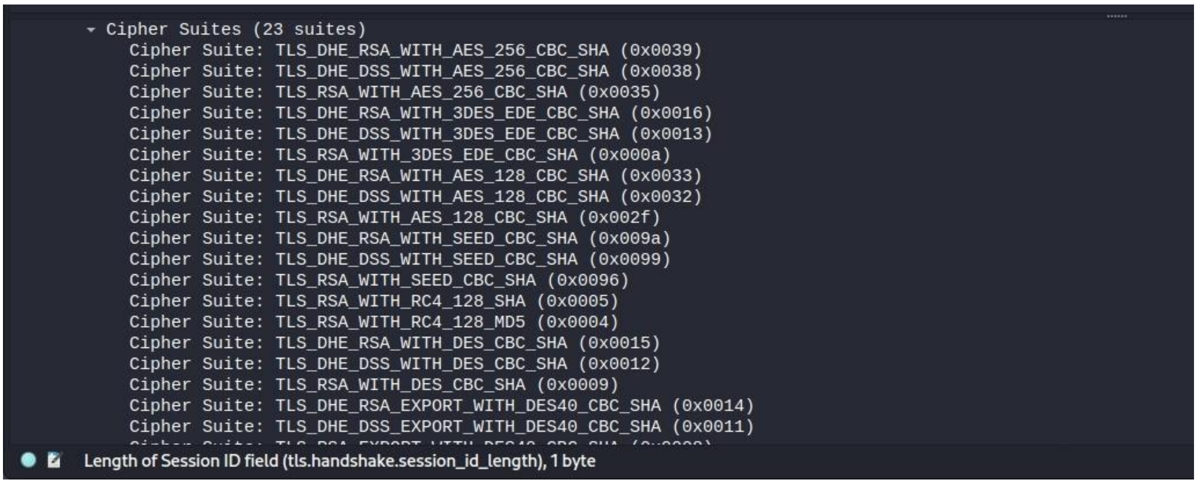
```
▸ Transmission Control Protocol, Src Port: 60245, Dst Port: 443, Seq: 1, Ack: 1, Len: 120
▾ Transport Layer Security
  ▾ TLSv1 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 115
    ▾ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 111
        Version: TLS 1.0 (0x0301)
      ▾ Random: 501778d316c25064f7cb0209b336ab332d969b8e091d26d4ccd04b731d7e550f
          GMT Unix Time: Jul 31, 2012 11:48:59.000000000 IST
          Random Bytes: 16c25064f7cb0209b336ab332d969b8e091d26d4ccd04b731d7e550f
        Session ID Length: 0
        Cipher Suites Length: 46
      ▸ Cipher Suites (23 suites)
        Compression Methods Length: 2
      ▸ Compression Methods (2 methods)
```



```
▸ Frame 6: 1484 bytes on wire (11872 bits), 1484 bytes captured (11872 bits) on interface en0, id 0
▸ Ethernet II, Src: Cisco-Li_e3:e9:8d (00:16:b6:e3:e9:8d), Dst: Apple_a2:05:1d (70:56:81:a2:05:1d)
▸ Internet Protocol Version 4, Src: 173.194.79.106, Dst: 192.168.1.102
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 60245, Seq: 1, Ack: 121, Len: 1418
▾ Transport Layer Security
  ▾ TLSv1 Record Layer: Handshake Protocol: Server Hello
      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: 501778d3d52d556ed20e072f638f0a51e9724d66ef5f13769d3a52e00161a893
          GMT Unix Time: Jul 31, 2012 11:48:59.000000000 IST
          Random Bytes: d52d556ed20e072f638f0a51e9724d66ef5f13769d3a52e00161a893
        Session ID Length: 32
```

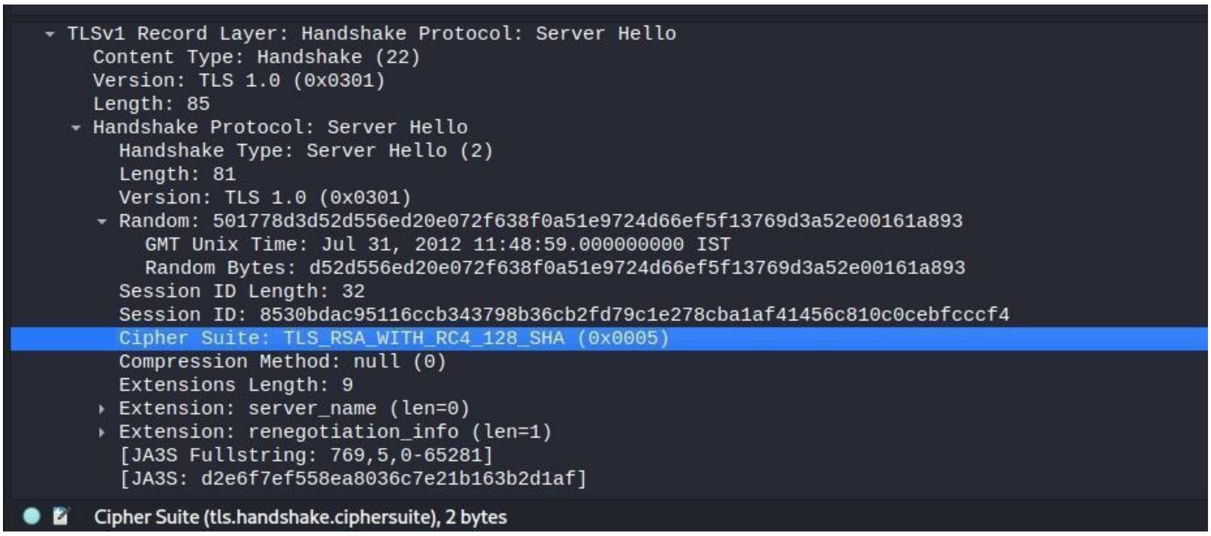
3. What Cipher suite is chosen by the Server? Give its name and value. The Client will list the different cipher methods it supports, and the Server will pick one of these methods to use.

Ans:



```
      ▾ Cipher Suites (23 suites)
          Cipher Suite: TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x0039)
          Cipher Suite: TLS_DHE_DSS_WITH_AES_256_CBC_SHA (0x0038)
          Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
          Cipher Suite: TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA (0x0016)
          Cipher Suite: TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (0x0013)
          Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
          Cipher Suite: TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x0033)
          Cipher Suite: TLS_DHE_DSS_WITH_AES_128_CBC_SHA (0x0032)
          Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
          Cipher Suite: TLS_DHE_RSA_WITH_SEED_CBC_SHA (0x009a)
          Cipher Suite: TLS_DHE_DSS_WITH_SEED_CBC_SHA (0x0099)
          Cipher Suite: TLS_RSA_WITH_SEED_CBC_SHA (0x0096)
          Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
          Cipher Suite: TLS_RSA_WITH_RC4_128_MD5 (0x0004)
          Cipher Suite: TLS_DHE_RSA_WITH_DES_CBC_SHA (0x0015)
          Cipher Suite: TLS_DHE_DSS_WITH_DES_CBC_SHA (0x0012)
          Cipher Suite: TLS_RSA_WITH_DES_CBC_SHA (0x0009)
          Cipher Suite: TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA (0x0014)
          Cipher Suite: TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA (0x0011)
          Cipher Suite: TLS_RSA_EXPORT_WITH_DES40_CBC_SHA (0x0008)
```

Length of Session ID field (tls.handshake.session_id_length), 1 byte



```
▾ TLSv1 Record Layer: Handshake Protocol: Server Hello
    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: 501778d3d52d556ed20e072f638f0a51e9724d66ef5f13769d3a52e00161a893
        GMT Unix Time: Jul 31, 2012 11:48:59.000000000 IST
        Random Bytes: d52d556ed20e072f638f0a51e9724d66ef5f13769d3a52e00161a893
      Session ID Length: 32
      Session ID: 8530bdac95116ccb343798b36cb2fd79c1e278cba1af41456c810c0cebfcccf4
      Cipher Suite: TLS_RSA_WITH_RC4_128_SHA (0x0005)
      Compression Method: null (0)
      Extensions Length: 9
    ▸ Extension: server_name (len=0)
    ▸ Extension: renegotiation_info (len=1)
      [JA3S Fullstring: 769,5,0-65281]
      [JA3S: d2e6f7ef558ea8036c7e21b163b2d1af]
```

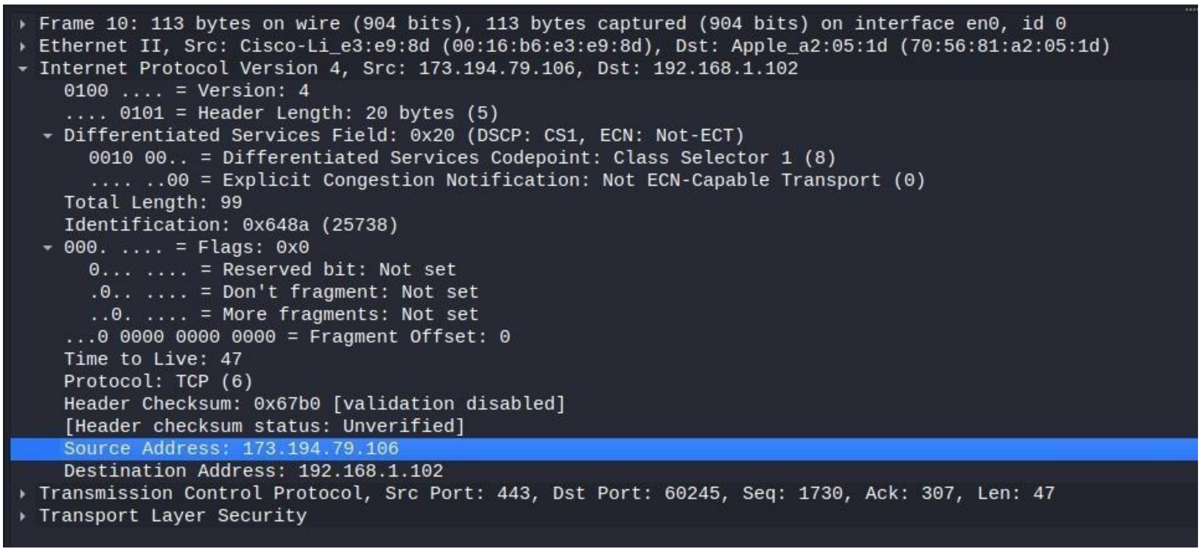
Cipher Suite (tls.handshake.ciphersuite), 2 bytes

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.

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.

Ans:Server to client



```
▸ Frame 10: 113 bytes on wire (904 bits), 113 bytes captured (904 bits) on interface en0, id 0
▸ Ethernet II, Src: Cisco-Li_e3:e9:8d (00:16:b6:e3:e9:8d), Dst: Apple_a2:05:1d (70:56:81:a2:05:1d)
▾ Internet Protocol Version 4, Src: 173.194.79.106, Dst: 192.168.1.102
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▾ Differentiated Services Field: 0x20 (DSCP: CS1, ECN: Not-ECT)
      0010 00.. = Differentiated Services Codepoint: Class Selector 1 (8)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 99
    Identification: 0x648a (25738)
  ▾ 000. .... = Flags: 0x0
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 47
    Protocol: TCP (6)
    Header Checksum: 0x67b0 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 173.194.79.106
    Destination Address: 192.168.1.102
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 60245, Seq: 1730, Ack: 307, Len: 47
▸ Transport Layer Security
```

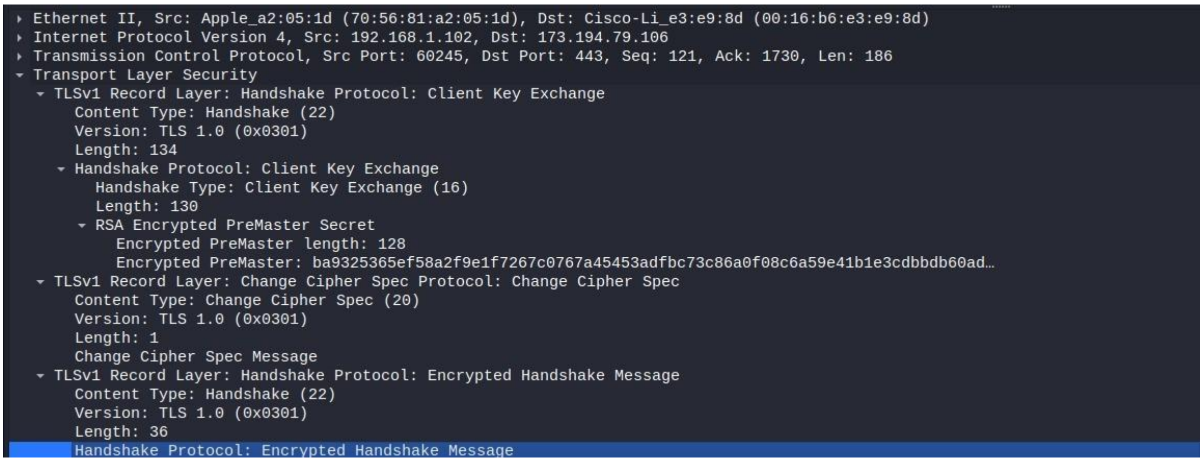
A Certificate message will contain one or more certificates, as needed for one party to verify the identity of the other party from its roots of trust certificates. You can inspect those certificates in your browser.

Client Key Exchange and Change Cipher Messages

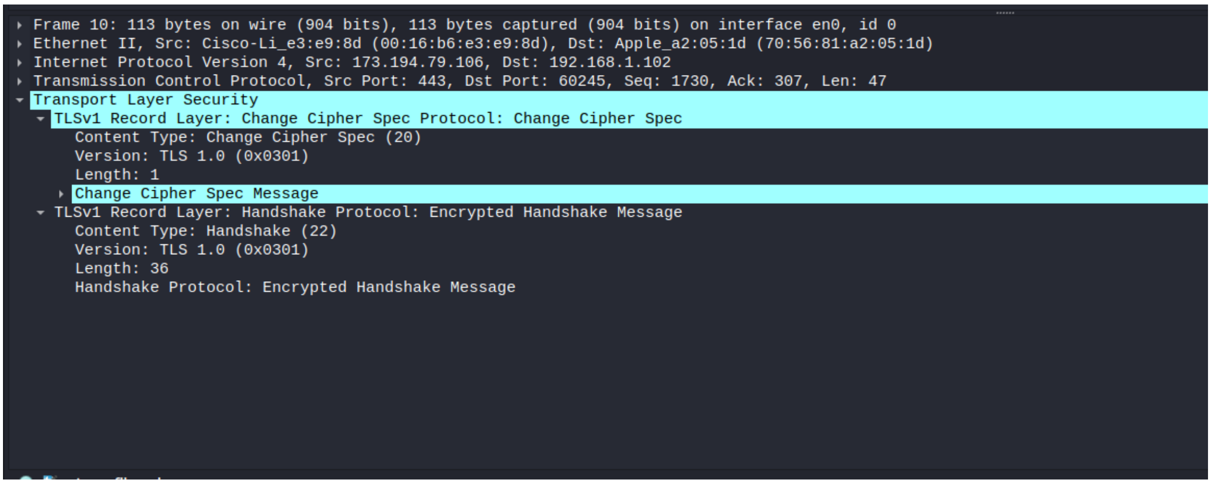
Find and inspect the details of the Client Key Exchange and Change Cipher messages, expanding their various details. The key exchange message is sent to pass keying information so that both sides will have the same secret session key. The change cipher message signal a switch to a new encryption scheme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

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

Ans:BOTH



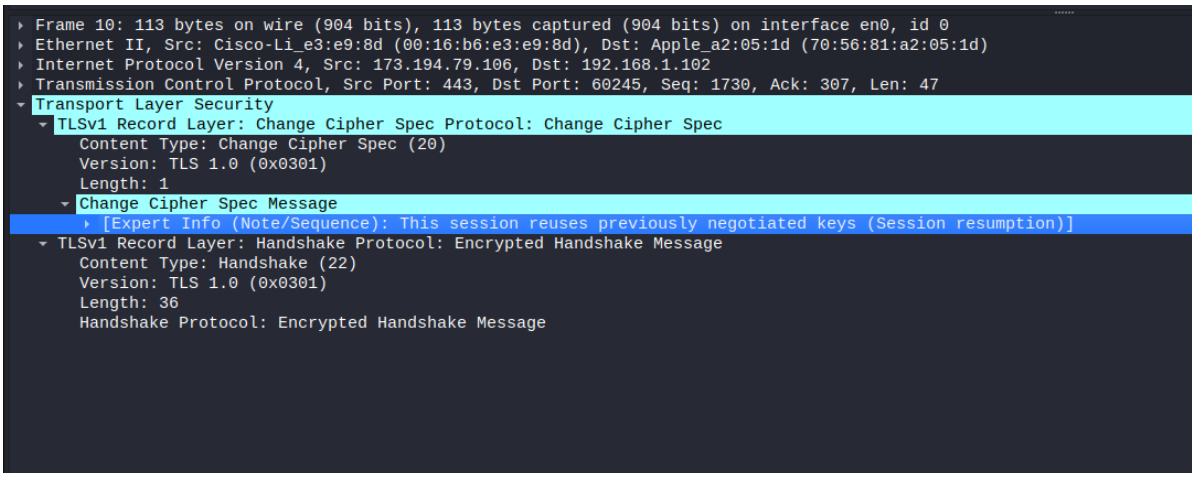
```
Ethernet II, Src: Apple_a2:05:1d (70:56:81:a2:05:1d), Dst: Cisco-Li_e3:e9:8d (00:16:b6:e3:e9:8d)
Internet Protocol Version 4, Src: 192.168.1.102, Dst: 173.194.79.106
Transmission Control Protocol, Src Port: 60245, Dst Port: 443, Seq: 121, Ack: 1730, Len: 186
Transport Layer Security
  TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 134
    Handshake Protocol: Client Key Exchange
      Handshake Type: Client Key Exchange (16)
      Length: 130
      RSA Encrypted PreMaster Secret
        Encrypted PreMaster length: 128
        Encrypted PreMaster: ba9325365ef58a2f9e1f7267c0767a45453adfbc73c86a0f08c6a59e41b1e3cdbbdb60ad…
  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
  TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 36
    Handshake Protocol: Encrypted Handshake Message
```



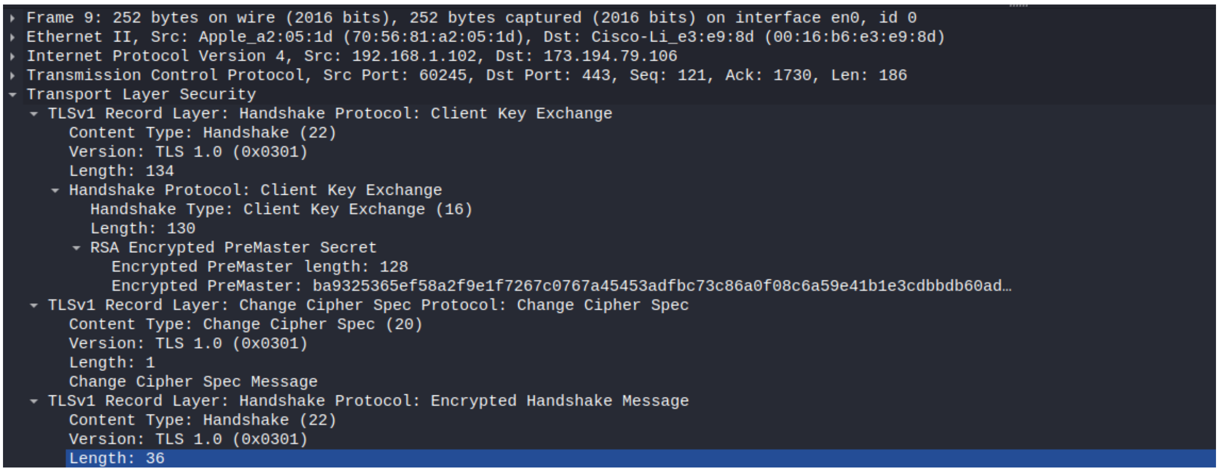
```
Frame 10: 113 bytes on wire (904 bits), 113 bytes captured (904 bits) on interface en0, id 0
Ethernet II, Src: Cisco-Li_e3:e9:8d (00:16:b6:e3:e9:8d), Dst: Apple_a2:05:1d (70:56:81:a2:05:1d)
Internet Protocol Version 4, Src: 173.194.79.106, Dst: 192.168.1.102
Transmission Control Protocol, Src Port: 443, Dst Port: 60245, Seq: 1730, Ack: 307, Len: 47
Transport Layer Security
  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
  TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 36
    Handshake Protocol: Encrypted Handshake Message
```

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

Ans:



```
Frame 10: 113 bytes on wire (904 bits), 113 bytes captured (904 bits) on interface en0, id 0
Ethernet II, Src: Cisco-Li_e3:e9:8d (00:16:b6:e3:e9:8d), Dst: Apple_a2:05:1d (70:56:81:a2:05:1d)
Internet Protocol Version 4, Src: 173.194.79.106, Dst: 192.168.1.102
Transmission Control Protocol, Src Port: 443, Dst Port: 60245, Seq: 1730, Ack: 307, Len: 47
Transport Layer Security
  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
      [Expert Info (Note/Sequence): This session reuses previously negotiated keys (Session resumption)]
  TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
    Content Type: Handshake (22)
    Version: TLS 1.0 (0x0301)
    Length: 36
    Handshake Protocol: Encrypted Handshake Message
```



```
▸ Frame 9: 252 bytes on wire (2016 bits), 252 bytes captured (2016 bits) on interface en0, id 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: 121, Ack: 1730, Len: 186
▾ Transport Layer Security
  ▾ TLSv1 Record Layer: Handshake Protocol: Client Key Exchange
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 134
    ▾ Handshake Protocol: Client Key Exchange
        Handshake Type: Client Key Exchange (16)
        Length: 130
      ▾ RSA Encrypted PreMaster Secret
          Encrypted PreMaster length: 128
          Encrypted PreMaster: ba9325365ef58a2f9e1f7267c0767a45453adfbc73c86a0f08c6a59e41b1e3cdbbdb60ad…
  ▾ 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
  ▾ TLSv1 Record Layer: Handshake Protocol: Encrypted Handshake Message
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 36
```

Conclusion:

Performed the experiment successfully.

Wireshark is used to analyse the packets of various protocols such as TCP, UDP, SSL, TLS, etc.