TLS

Yogesh P
201EE138

2. A first look at the captured trace

1.What is the packet number in your trace that contains the initial TCP SYN message? (By “packet number,” we meant the number in the “No.” column at the left of the Wireshark display, not the sequence number in the TCP segment itself).

A. 17

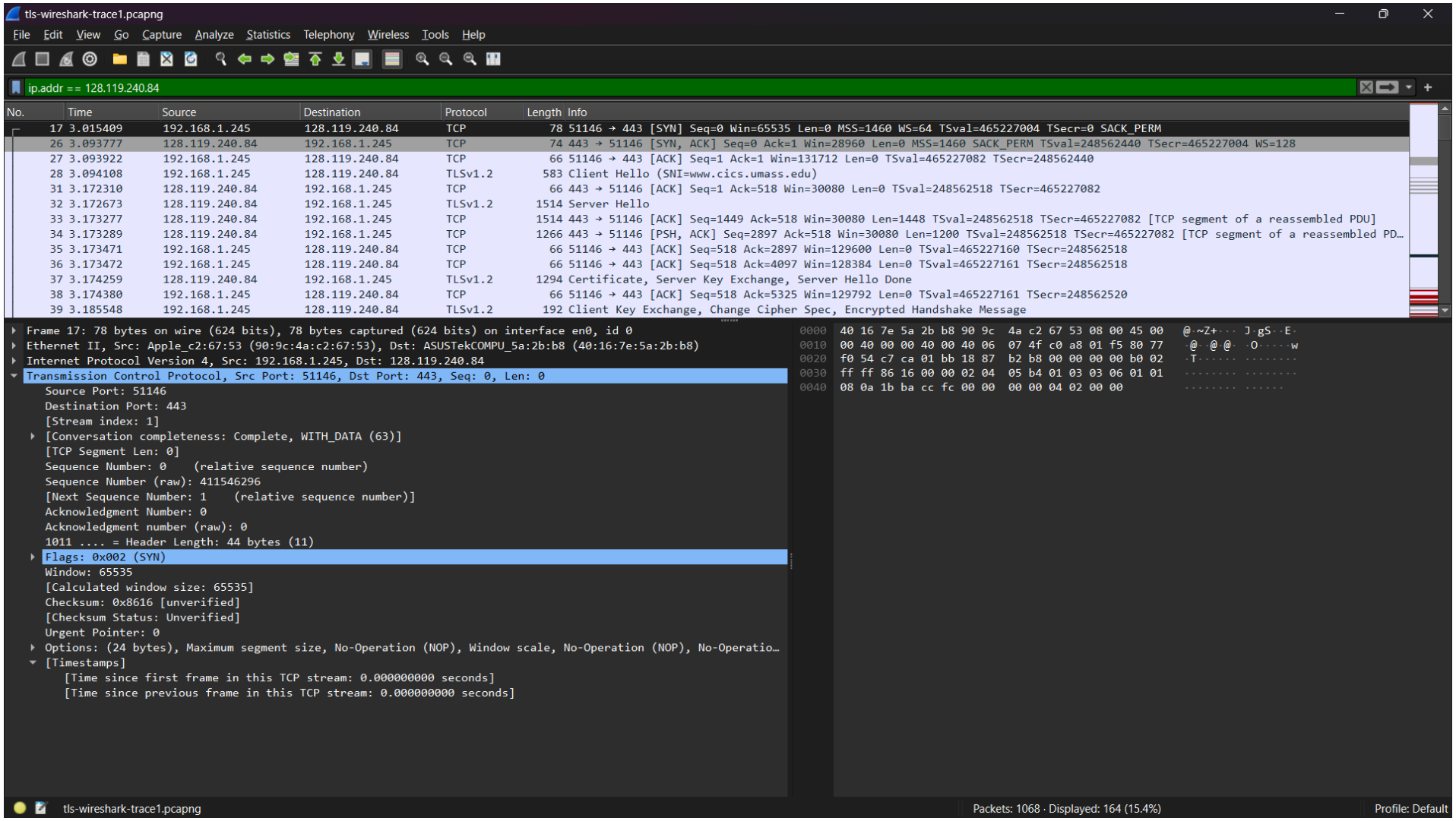
2.Is the TCP connection set up before or after the first TLS message is sent from client to server?

A. After

3. What is the packet number in your trace that contains the TLS *Client Hello* message?

A. 28

3. The TLS Handshake: Client Hello message


tls-wireshark-trace1.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 128.119.240.84
No. Time Source Destination Protocol Length Info
17 3.015409 192.168.1.245 128.119.240.84 TCP 78 51146 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=465227004 TSecr=0 SACK_PERM
26 3.093777 128.119.240.84 192.168.1.245 TCP 74 443 → 51146 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM TSval=248562440 TSecr=465227004 WS=128
27 3.093922 192.168.1.245 128.119.240.84 TCP 66 51146 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=465227082 TSecr=248562440
28 3.094108 192.168.1.245 128.119.240.84 TLSv1.2 583 Client Hello (SNI=www.cics.umass.edu)
31 3.172310 128.119.240.84 192.168.1.245 TCP 66 443 → 51146 [ACK] Seq=1 Ack=518 Win=30080 Len=0 TSval=248562518 TSecr=465227082
32 3.172673 128.119.240.84 192.168.1.245 TLSv1.2 1514 Server Hello
33 3.173277 128.119.240.84 192.168.1.245 TCP 1514 443 → 51146 [ACK] Seq=1449 Ack=518 Win=30080 Len=1448 TSval=248562518 TSecr=465227082 [TCP segment of a reassembled PDU]
34 3.173289 128.119.240.84 192.168.1.245 TCP 1266 443 → 51146 [PSH, ACK] Seq=2897 Ack=518 Win=30080 Len=1200 TSval=248562518 TSecr=465227082 [TCP segment of a reassembled PD…
35 3.173471 192.168.1.245 128.119.240.84 TCP 66 51146 → 443 [ACK] Seq=518 Ack=2897 Win=129600 Len=0 TSval=465227160 TSecr=248562518
36 3.173472 192.168.1.245 128.119.240.84 TCP 66 51146 → 443 [ACK] Seq=518 Ack=4097 Win=128384 Len=0 TSval=465227161 TSecr=248562518
37 3.174259 128.119.240.84 192.168.1.245 TLSv1.2 1294 Certificate, Server Key Exchange, Server Hello Done
38 3.174380 192.168.1.245 128.119.240.84 TCP 66 51146 → 443 [ACK] Seq=518 Ack=5325 Win=129792 Len=0 TSval=465227161 TSecr=248562520
39 3.185548 192.168.1.245 128.119.240.84 TLSv1.2 192 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
Frame 17: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface en0, id 0
Ethernet II, Src: Apple_c2:67:53 (90:9c:4a:c2:67:53), Dst: ASUSTekCOMPU_5a:2b:b8 (40:16:7e:5a:2b:b8)
Internet Protocol Version 4, Src: 192.168.1.245, Dst: 128.119.240.84
Transmission Control Protocol, Src Port: 51146, Dst Port: 443, Seq: 0, Len: 0
Source Port: 51146
Destination Port: 443
[Stream index: 1]
[Conversation completeness: Complete, WITH_DATA (63)]
[TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 411546296
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
1011 .... = Header Length: 44 bytes (11)
Flags: 0x002 (SYN)
Window: 65535
[Calculated window size: 65535]
Checksum: 0x8616 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
Options: (24 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operatio…
[Timestamps]
[Time since first frame in this TCP stream: 0.000000000 seconds]
[Time since previous frame in this TCP stream: 0.000000000 seconds]
0000 40 16 7e 5a 2b b8 90 9c 4a c2 67 53 08 00 45 00
0010 00 40 00 00 40 00 40 06 07 4f c0 a8 01 f5 80 77
0020 f0 54 c7 ca 01 bb 18 87 b2 b8 00 00 00 00 b0 02
0030 ff ff 86 16 00 00 02 04 05 b4 01 03 03 06 01 01
0040 08 0a 1b ba cc fc 00 00 00 00 04 02 00 00
tls-wireshark-trace1.pcapng
Packets: 1068 · Displayed: 164 (15.4%)
Profile: Default

4.What version of TLS is your client running, as declared in the *Client Hello* message?

A. TLSv1.2

5.How many cipher suites are supported by your client, as declared in the *Client Hello* message? A cipher suite is a set of related cryptographic algorithms that determine how session keys will be derived, and how data will be encrypted and be digitally signed via a HMAC algorithm.

A. 34 bytes/2 =17 cipher suites.

6.Your client generates and sends a string of “random bytes” to the server in the *Client Hello* message. What are the first two hexadecimal digits in the random bytes field of the *Client Hello* message? Enter the two hexadecimal digits

A.4b

7.What is the purpose(s) of the “random bytes” field in the *Client Hello* message?

A. The random bytes field in the Client Hello message provides entropy for key generation, ensuring cryptographic strength, and includes version information for protocol negotiation, enhancing compatibility between client and server.

8.What is the packet number in your trace that contains the TLS *Server Hello* message?

A.32

3. The TLS Handshake: Server Hello message

9.Which cipher suite has been chosen by the server from among those offered in the earlier *Client Hello* message?

A. Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)

10.Does the *Server Hello* message contain random bytes, similar to how the *Client Hello* message contained random bytes? And if so, what is/are their purpose(s)?

A. Yes, The random bytes in the Server Hello message contribute to key generation, aid in version negotiation, and facilitate session resumption in TLS handshake.

11.What is the packet number in your trace for the TLS message part that contains the public key certificate for the www.cics.umass.edu server (actually the www.cs.umass.edu server)?

A.37

12.A server may return more than one certificate. If more than one certificate is returned, are all of these certificates for www.cs.umass.edu? If not all are for www.cs.umass.edu, then who *are* these other certificates for? You can determine who the certificate is for by checking the id-at-commonName field in the retuned certificate.

A.No

13.What is the name of the certification authority that issued the certificate for id-at-commonName=www.cs.umass.edu?

A. Common Name (CN)=InCommon RSA Server CA, Organization=(O)Internet2

14.What digital signature algorithm is used by the CA to sign this certificate?

A.PKCS #1 SHA-384 With RSA Encryption

15.Let's take a look at what a real public key looks like! What are the first four hexadecimal digits of the modulus of the public key being used by www.cics.umass.edu? Enter the four hexadecimal digits (without spaces between the hex digits and without any leading '0x' , using lowercase letters where needed, and including any leading 0s after '0x'). Hint: this information can be found in subjectPublicKeyInfo subfield of the SignedCertificate field of the certificate for www.cs.umass.edu.

A. 5C D4

16.Look in your trace to find messages between the client and a CA to get the CA's public key information, so that the client can verify that the CA-signed certificate sent by the server is indeed valid and has not been forged or altered. Do you see such message in your trace? If so, what is the number in the trace of the first packet sent from your client to the CA? If not, explain why the client did not contact the CA.

A.Yes , 8daf346bc54adc94fd7762e53535c6597e4c5e77a94a28a7e7eb961d92bb81d7

17.What is the packet number in your trace for the TLS message part that contains the *Server Hello Done* TLS record?

A.37

4. The TLS Handshake: wrapping up the handshake

18.What is the packet number in your trace for the TLS message that contains the public key information, *Change Cipher Spec,* and *Encrypted Handshake* message, being sent from client to server?

A.39

19.Does the client provide its own CA-signed public key certificate back to the server? If so, what is the packet number in your trace containing your client's certificate?

A: yes, 39

5. Application data

20.What symmetric key cryptography algorithm is being used by the client and server to encrypt application data (in this case, HTTP messages)?

A.AES (Advanced Encryption Standard)

21.In which of the TLS messages is this symmetric key cryptography algorithm finally decided and declared?

A.Change Cipher Spec

22.What is the packet number in your trace for the first encrypted message carrying application data from client to server?

A.41

23.What do you think the content of this encrypted application-data is, given that this trace was generated by fetching the homepage of www.cics.umass.edu?

A.contains the HTML content of the homepage.

24.What packet number contains the client-to-server TLS message that shuts down the TLS connection? Because TLS messages are encrypted in our Wireshark traces, we can't actually look *inside* a TLS message and so we'll have to make an educated guess here.

A.358, 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.