

CS315 : Computer Networks Lab

Assignment 13

Sourabh Bhosale (200010004)

April 11, 2023

1 Part-1: Capturing packets in an TLS session

No.	Time	Source	Destination	Protocol	Length	Info
935	5.776728	10.196.77.134	128.119.240.84	TCP	78	51196 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=27991149 TSecr=0 SACK_PERM
936	5.777612	10.196.77.134	128.119.240.84	TCP	78	51197 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2782008813 TSecr=0 SACK_PERM
937	5.779840	128.119.240.84	10.196.77.134	TCP	74	443 → 51196 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM TSval=428780840 TSecr=2799
938	5.779939	10.196.77.134	128.119.240.84	TCP	66	51196 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=27991152 TSecr=428780840
939	5.780385	10.196.77.134	128.119.240.84	TLSv1.2	583	Client Hello
940	5.780963	128.119.240.84	10.196.77.134	TCP	74	443 → 51197 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM TSval=428780840 TSecr=2782
941	5.781851	10.196.77.134	128.119.240.84	TCP	66	51197 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=2782008817 TSecr=428780840
942	5.781378	10.196.77.134	128.119.240.84	TLSv1.2	583	Client Hello
943	5.783116	128.119.240.84	10.196.77.134	TCP	66	443 → 51196 [ACK] Seq=1 Ack=518 Win=15616 Len=0 TSval=428780840 TSecr=27991152
944	5.784885	128.119.240.84	10.196.77.134	TCP	66	443 → 51197 [ACK] Seq=1 Ack=518 Win=15616 Len=0 TSval=428780840 TSecr=2782008817
1046	6.357667	128.119.240.84	10.196.77.134	TLSv1.2	1514	Server Hello
1047	6.357687	128.119.240.84	10.196.77.134	TCP	1514	443 → 51197 [PSH, ACK] Seq=1449 Ack=518 Win=15616 Len=1448 TSval=428780889 TSecr=2782008817 [TCP s
1048	6.357866	10.196.77.134	128.119.240.84	TCP	66	51197 → 443 [ACK] Seq=518 Ack=2897 Win=128832 Len=0 TSval=2782009393 TSecr=428780889
1049	6.359811	128.119.240.84	10.196.77.134	TCP	1266	443 → 51197 [PSH, ACK] Seq=2897 Ack=518 Win=15616 Len=1200 TSval=428780889 TSecr=2782008817 [TCP
1050	6.359813	128.119.240.84	10.196.77.134	TLSv1.2	1289	Certificate, Server Key Exchange, Server Hello Done
1051	6.359814	128.119.240.84	10.196.77.134	TLSv1.2	1514	Server Hello
1052	6.359815	128.119.240.84	10.196.77.134	TCP	1514	443 → 51196 [PSH, ACK] Seq=1449 Ack=518 Win=15616 Len=1448 TSval=428780889 TSecr=27991152 [TCP s
1053	6.359816	128.119.240.84	10.196.77.134	TCP	1266	443 → 51196 [PSH, ACK] Seq=2897 Ack=518 Win=15616 Len=1200 TSval=428780889 TSecr=27991152 [TCP s
1054	6.359817	128.119.240.84	10.196.77.134	TLSv1.2	1289	Certificate, Server Key Exchange, Server Hello Done
1055	6.359150	10.196.77.134	128.119.240.84	TCP	66	51197 → 443 [ACK] Seq=518 Ack=5320 Win=128640 Len=0 TSval=2782009395 TSecr=428780889
1056	6.359213	10.196.77.134	128.119.240.84	TCP	66	51196 → 443 [ACK] Seq=518 Ack=5320 Win=126400 Len=0 TSval=27991732 TSecr=428780890
1057	6.359770	10.196.77.134	128.119.240.84	TCP	66	[TCP Window Update] 51196 → 443 [ACK] Seq=518 Ack=5320 Win=131072 Len=0 TSval=27991732 TSecr=428
1058	6.365086	10.196.77.134	128.119.240.84	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
1059	6.366038	10.196.77.134	128.119.240.84	TLSv1.2	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
1060	6.368240	128.119.240.84	10.196.77.134	TCP	66	443 → 51197 [ACK] Seq=5320 Ack=644 Win=15616 Len=0 TSval=428780899 TSecr=2782009401
1061	6.368240	128.119.240.84	10.196.77.134	TCP	66	443 → 51196 [ACK] Seq=5320 Ack=644 Win=15616 Len=0 TSval=428780899 TSecr=27991738
1115	6.663111	128.119.240.84	10.196.77.134	TLSv1.2	340	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message

> Frame 935: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface en0, id 0
> Ethernet II, Src: Apple_Ad:cf:45 (14:7d:da:ad:cf:45), Dst: ExtremeNetworks_9a:82:e8 (02:04:96:9a:82:e8)
> Internet Protocol Version 4, Src: 10.196.77.134, Dst: 128.119.240.84
> Transmission Control Protocol, Src Port: 51196, Dst Port: 443, Seq: 0, Len: 0

** was unexpected in this context.

Packets: 8840 · Displayed: 2459 (27.8%) · Dropped: 0 (0.0%) Profile: Default

2 Part-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)

In my case, that is packet number 935.

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

TCP connection is set up before the first TLS message is sent.

3 Part-3: The TLS Handshake: Client Hello message

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

Packet number : 939

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

TLSv1.2

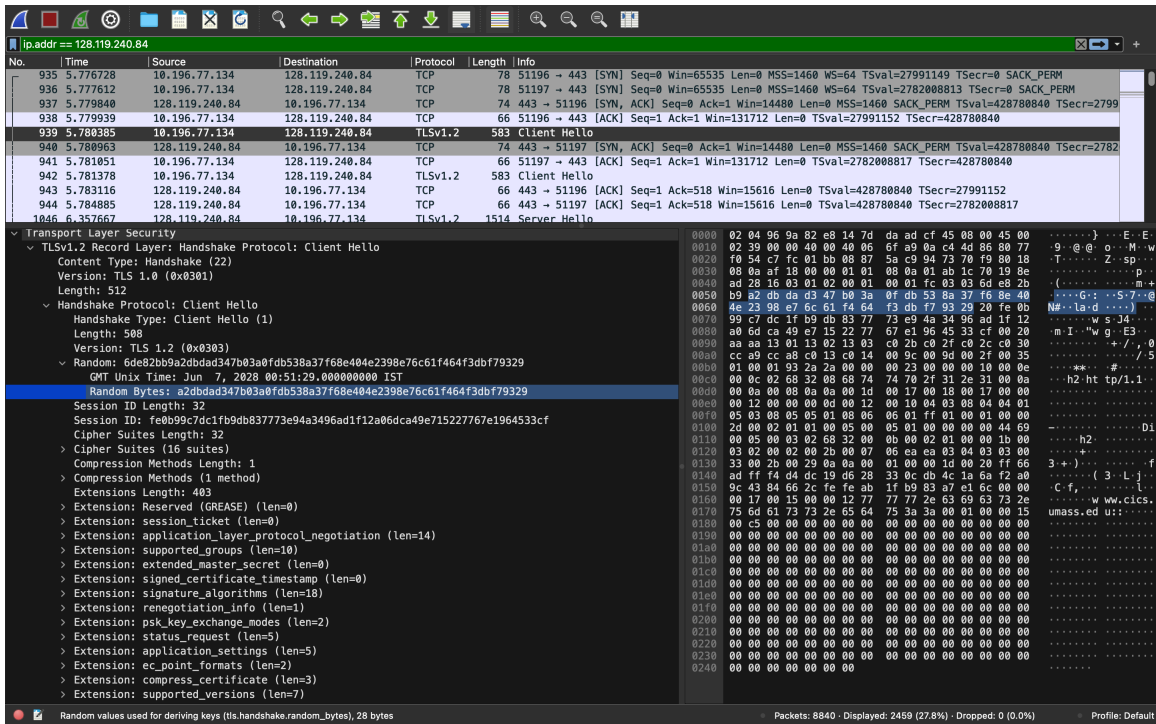
3. How many cipher suites are supported by your client, as declared in the Client Hello message? A cipher suite is a set of related cryptographic algorithms that determine how session keys will be derived, and hoid-at-commonName=www.cs.umass.edua HMAC algorithm.

16 suites

4. 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 (without spaces between the hex digits and without any leading ‘ox’, using lowercase letters where needed). Hint: be careful to fully dig into the Random field to find the Random Bytes subfield (do not consider the GMT UNIX Time subfield of Random).

First two hexadecimal digits in the random bytes field : a2

Random Bytes: a2dbdad347b03aofdb538a37f68e404e2398e76c61f464f3dbf79329



5. What is the purpose(s) of the “random bytes” field in the Client Hello message? Note: you’ll have to do some searching and reading to get the answer to this question; see section 8.6 and in RFC 5246 (section 8.1 in RFC 5246 in particular).

When resuming a session, the same master key is used to generate key block. So use of client and server random bytes ensures that key block will be different in every handshake. "Random bytes" field is used to prevent replay attacks on the network.

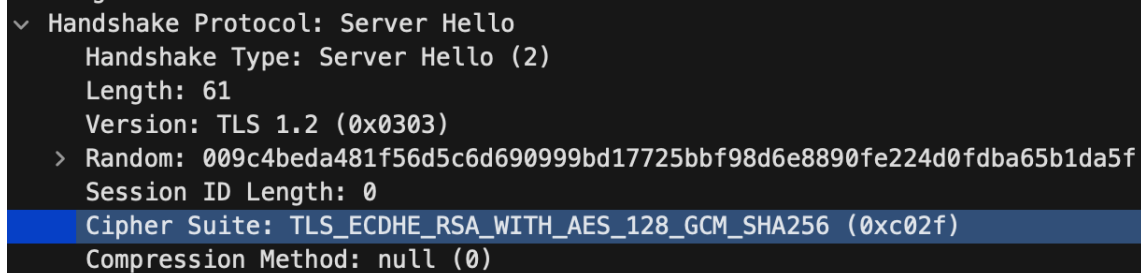
4 Part-4: The TLS Handshake: Server Hello message

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

Packet number : 1046

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

Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)



```

  ▾ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 61
    Version: TLS 1.2 (0x0303)
    > Random: 009c4beda481f56d5c6d690999bd17725bbf98d6e8890fe224d0fdb65b1da5f
    Session ID Length: 0
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
    Compression Method: null (0)

```

3. 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)?

Yes, the Server Hello message contain random bytes. When resuming a session, the same master key is used to generate key block. So use of client and server random bytes ensures that key block will be different in every handshake. "Random bytes" field is used to prevent replay attacks on the network.

4. 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)?

Packet number : 1050

5. 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 returned certificate.

Yes, both the the certificates are for www.cs.umass.edu. I could determine that by checking the field 'id-at-commonName'. Refer below two figures for the same.

```

1049 6.359811 128.119.240.84 10.196.77.134 TCP 1266 443 → 51197 [PSH, ACK] Seq=2897 Ack=518 Win=15616 Len=1200 TSval=428780889 TSecr=2782008817 [TCP s
1050 6.359813 128.119.240.84 10.196.77.134 TLSv1.2 1289 Certificate, Server Key Exchange, Server Hello Done
1051 6.359814 128.119.240.84 10.196.77.134 TLSv1.2 1514 Server Hello
1052 6.359815 128.119.240.84 10.196.77.134 TCP 1514 443 → 51196 [PSH, ACK] Seq=1449 Ack=518 Win=15616 Len=1448 TSval=428780889 TSecr=279911152 [TCP s
1053 6.359816 128.119.240.84 10.196.77.134 TCP 1266 443 → 51196 [PSH, ACK] Seq=2897 Ack=518 Win=15616 Len=1200 TSval=428780889 TSecr=279911152 [TCP s
1054 6.359817 128.119.240.84 10.196.77.134 TLSv1.2 1289 Certificate, Server Key Exchange, Server Hello Done
1055 6.359150 10.196.77.134 128.119.240.84 TCP 66 51197 → 443 [ACK] Seq=518 Ack=5320 Win=128640 Len=0 TSval=2782009395 TSecr=428780889

Version: TLS 1.2 (0x0303)
Length: 4897
  Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 4893
    Certificates Length: 4890
    Certificates (4890 bytes)
      Certificate Length: 1842
      > Certificate: 3082072e30820616a00302010202103090854915311cde05eb63eb08727271300d06092a... (id-at-commonName=www.cs.umass.edu,id-at-organizationName=University of Massa...
      Certificate Length: 1533
      > Certificate: 308205f9308203e1a00302010202104720d0fa85461a7e17a1640291846374300d06092a... (id-at-commonName=InCommon RSA Server CA,id-at-organizationalUnitName=InCommon...
      Certificate Length: 1506
      > Certificate: 308205de308203c6a003020102021001fd6d30fca3ca51a81bbc640e35032d300d06092a... (id-at-commonName=USERTrust RSA Certification Authority,id-at-organizationName=...
    Transport Layer Security
      TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange

```

```

1049 6.359811 128.119.240.84 10.196.77.134 TCP 1266 443 → 51197 [PSH, ACK] Seq=2897 Ack=518 Win=15616 Len=1200 TSval=428780889 TSecr=2782008817 [TCP s
1050 6.359813 128.119.240.84 10.196.77.134 TLSv1.2 1289 Certificate, Server Key Exchange, Server Hello Done
1051 6.359814 128.119.240.84 10.196.77.134 TLSv1.2 1514 Server Hello
1052 6.359815 128.119.240.84 10.196.77.134 TCP 1514 443 → 51196 [PSH, ACK] Seq=1449 Ack=518 Win=15616 Len=1448 TSval=428780889 TSecr=279911152 [TCP s
1053 6.359816 128.119.240.84 10.196.77.134 TCP 1266 443 → 51196 [PSH, ACK] Seq=2897 Ack=518 Win=15616 Len=1200 TSval=428780889 TSecr=279911152 [TCP s
1054 6.359817 128.119.240.84 10.196.77.134 TLSv1.2 1289 Certificate, Server Key Exchange, Server Hello Done
1055 6.359150 10.196.77.134 128.119.240.84 TCP 66 51197 → 443 [ACK] Seq=518 Ack=5320 Win=128640 Len=0 TSval=2782009395 TSecr=428780889

Handshake Protocol: Certificate
  Handshake Type: Certificate (11)
  Length: 4893
  Certificates Length: 4890
  Certificates (4890 bytes)
    Certificate Length: 1842
    > Certificate: 3082072e30820616a00302010202103090854915311cde05eb63eb08727271300d06092a... (id-at-commonName=www.cs.umass.edu,id-at-organizationName=University of Massachu...
    Certificate Length: 1533
    > Certificate: 308205f9308203e1a00302010202104720d0fa85461a7e17a1640291846374300d06092a... (id-at-commonName=InCommon RSA Server CA,id-at-organizationalUnitName=InCommon...
    Certificate Length: 1506
    > Certificate: 308205de308203c6a003020102021001fd6d30fca3ca51a81bbc640e35032d300d06092a... (id-at-commonName=USERTrust RSA Certification Authority,id-at-organizationName=...
  Transport Layer Security
    TLSv1.2 Record Layer: Handshake Protocol: Server Key Exchange
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)

```

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

USERTrust RSA Certification Authority

7. What digital signature algorithm is used by the CA to sign this certificate? Hint: this information can be found in the signature subfield of the SignedCertificate field of the certificate for www.cs.umass.edu.

Algorithm Id: 1.2.840.113549.1.1.11 (sha256WithRSAEncryption)

8. 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.

First four hexadecimal digits : 3082, for full public key refer figure.

```

> signature (sha256WithRSAEncryption)
> issuer: rdnSequence (0)
> validity
> subject: rdnSequence (0)
> subjectPublicKeyInfo
  > algorithm (rsaEncryption)
    > subjectPublicKey: 3082010a0282010100b39e7296158da80176a2f1035c7c61f06120f9852aad0d20d4931a...
      modulus: 6x00b39e7296158da80176a2f1035c7c61f06120f9852aad0d20d4931a30842fec1b0724...
      publicExponent: 65537
  > extensions: 10 items
  > algorithmIdentifier (sha256WithRSAEncryption)
    Padding: 0
    encrypted: 798cald2fd37cc02bdc0a8e491326ea6ffad9ec7eb76ffccce9b7e90b4ae31c10e981492...
    Certificate Length: 1533
  > Certificate: 308205f9308203e1a003020104720d0fa85461a7e17a1640291846374300d06092a... (id-at-commonName
    Certificate Length: 1506
  > Certificate: 308205dc308203c6a003020102021001f6d60fca3c51a081bhc640c35032d300d06092a... (id-at-commonName

```

9. 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 a 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.

No, there doesn't seem to be such a message. It is possible that the client has cached the required information, and hence did not contact the CA.

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

Packet number : 1050 and 1054

1046	6.357667	128.119.240.84	10.196.77.134	TLSv1.2	1514	Server Hello
1050	6.359013	128.119.240.84	10.196.77.134	TLSv1.2	1289	Certificate, Server Key Exchange, Server Hello Done
1051	6.359014	128.119.240.84	10.196.77.134	TLSv1.2	1514	Server Hello
1054	6.359017	128.119.240.84	10.196.77.134	TLSv1.2	1289	Certificate, Server Key Exchange, Server Hello Done

5 Part-5: The TLS Handshake: wrapping up the handshake

1. 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?

Packet number : 1058

2. 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?

No, the client does not appear to provide its own CA-signed public key certificate back to the server. This could be because the goal of the authentication in this case is for the server's authenticity, the client is not obligated to authenticate.

6 Part-6: Application data

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

SHA-256 Algorithm is used with RSA Encryption by the client and server to encrypt application data.

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

This symmetric key cryptography algorithm was finally decided and declared in the TLS Server Hello Done message.

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

Packet number : 1119

4. 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?

The content of this field will be data transferred from client to server which will be encrypted for security reasons. Now also, Since that particular message is sent from client to the server, and HTTP is working on top of TLS, considering the TCP and TLS starting earlier (HTTP-over-TLS-over-TCP connection), this encrypted application data must contain the HTTP GET Request.

Actual content is, Encrypted Application Data: 0000000000000000126616ad104515a733f64518bob23953102

5. 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.

The Alert message is sent to signal a condition, such as notification that one party is closing the connection. (We can find for 'Encrypted Alert' packet in our trace.) 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.