Yug Patel
CS23MTECH14019

Assignment 6: Decrypting TLS and HTTP(S) using Wireshark++

Website: www.bankofamerica.com

Whenever possible, when answering the questions given below, you should produce a screenshot of the packet(s) within the trace that you used to answer the question asked. Highlight portions of the snapshot to explain your answer. To print a packet in wireshark GUI, use *File->Print Option*, choose *Selected packet only*, choose *Packet summary line,* and select the minimum amount of packet detail that you need to answer the question.

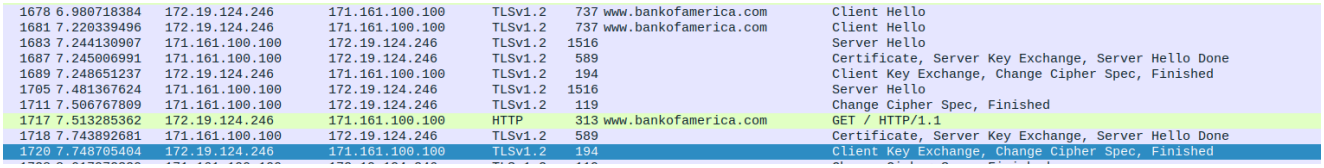
1. What browser did you use, what's the version number?
   Ans: User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:122.0)

2. List out various protocols that you noticed in the column named "Protocol" in the wireshark GUI from the time you keyed in the hostname of the bank in the browser till you start viewing application data. For each such protocol, mention its purpose in brief.
   Ans: DNS: To resolve the domain name
   TCP : For TCP Three-Way handshake, which provides process to process inorder reliable delivery of packets from sender to receiver.
   TLS: For TLS four phase handshake, required to share the secret attributes useful for secure communication by encrypting the channel/Pipe using the agreed upon things.
   OCSP: For checking the status of the certificate.
   HTTP: For making the request

3. Each of the TLS records begins with the same three fields (with possibly different values). One of these fields is "content type" and has a length of one byte. List all three fields and their lengths for the first 10 records in the trace.
   Ans:

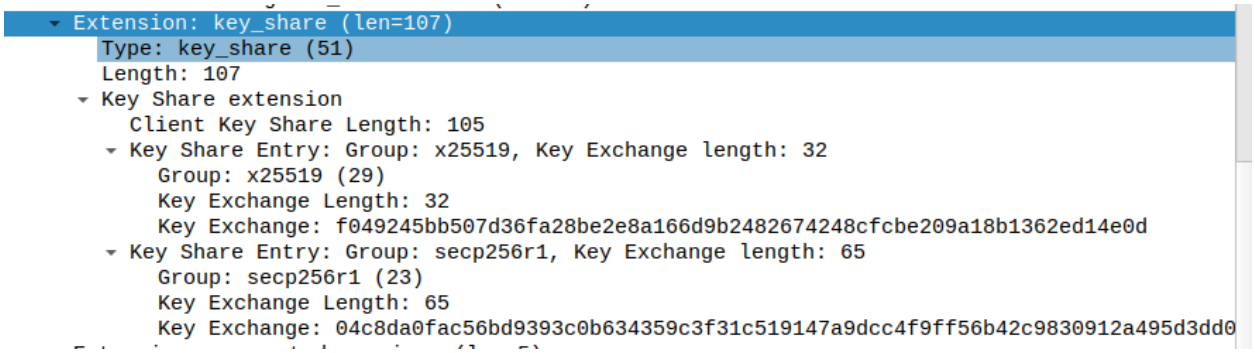
Packet Number	Content Type	Version	Length
1	Handshake (22)	TLS 1.0 (0x0301)	664
2	Handshake (22)	TLS 1.0 (0x0301)	664
3	Handshake (22)	TLS 1.2 (0x0303)	91
4	Handshake (22)	TLS 1.2 (0x0303)	4417
5	Handshake (22)	TLS 1.2 (0x0303)	70

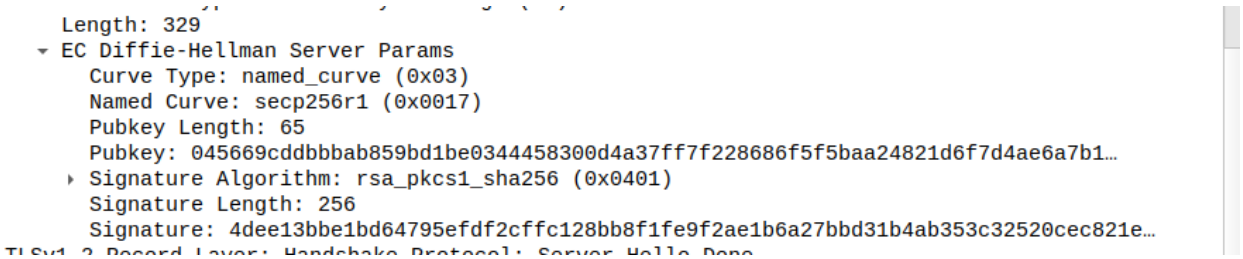
6	Handshake (22)	TLS 1.2 (0x0303)	91
7	Change Cipher Spec (20)	TLS 1.2 (0x0303)	1
8	Application Data (23)	TLS 1.2 (0x0303)	1664
9	Handshake (22)	TLS 1.2 (0x0303)	4417
10	Handshake (22)	TLS 1.2 (0x0303)	70



4. What are the key extensions that you noticed in the Client Hello message? By observing the Server Hello message, explain what extensions really used by the server for establishing TLS pipe?

   Ans: As shown in the below pictures, the client shares key extensions that it supports and out of which the server selects one and generates its key share.





5. Cipher Suites in ClientHello Record: Look at the first two and the last cipher suites offered by the client and compare them. What cipher suite the server selected?

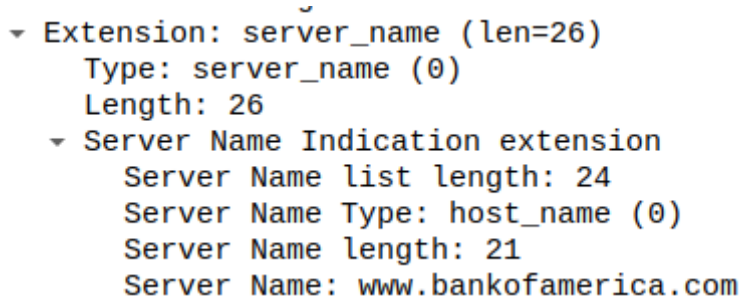
   Ans:

   Cipher Suite (First): TLS_AES_128_GCM_SHA256 (0x1301)
   Cipher Suite (Second): TLS_CHACHA20_POLY1305_SHA256 (0x1303)
   Cipher Suite (Last): TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)

1. TLS_AES_128_GCM_SHA256 (0x1301):
   - Key Exchange: Typically uses Elliptic Curve Diffie-Hellman (ECDHE) for key exchange.
   - Authentication: HMAC-SHA256 is used for data integrity.
   - Encryption: AES-128 in Galois/Counter Mode (GCM) provides authenticated encryption.
2. TLS_CHACHA20_POLY1305_SHA256 (0x1303):
   - Key Exchange: Often uses ECDHE for key exchange.
   - Authentication: HMAC-SHA256 is used for data integrity.
   - Encryption: ChaCha20 is used for encryption, and Poly1305 for authentication.
3. TLS_RSA_WITH_AES_256_CBC_SHA (0x0035):
   - Key Exchange: Uses RSA for key exchange.
   - Authentication: HMAC-SHA1 is used for data integrity (considered less secure than SHA256).
   - Encryption: AES-256 in Cipher Block Chaining (CBC) mode provides encryption.

6. What is the SNI value in ClientHello Record? What's its purpose? In other words, why is the client advertising it to the server?

   Ans: SNI stands for Server Name Indicator, this extension indicates the hostname that the client is attempting to connect to. It is helpful in cases when a single ip address is hosting multiple websites (Virtual Hosting) so that it responds with the appropriate certificate based on the requested hostname.



7. What is the ALPN value(s) in ClientHello Record? What's its purpose? Which one the server selected?
   Ans: ALPN (Application Layer Protocol Negotiation), this extension is used to negotiate the application protocol that will be used over the secured connection. The server does not express its preference for application layer protocol. In this case, the client and server will proceed with the TLS connection without negotiating a specific protocol for the application layer. The absence of ALPN negotiation may not be an issue in scenarios where the server is designed to handle multiple application-layer protocols or where the client can make an informed decision based on the context.

```
Extensions Length: 15
▸ Extension: renegotiation_info (len=1)
▸ Extension: ec_point_formats (len=2)
▸ Extension: extended_master_secret (len=0)
[JA3S Fullstring: 771,49199,65281-11-23]
[JA3S: 76c691f46143bf86e2d1bb73c6187767]
```

8. Does the ClientHello contain status_request, supported_versions, psk_key_exchange_modes extensions? If so, what do they convey to the server?
   Ans:

```
▾ Extension: status_request (len=5)
    Type: status_request (5)
    Length: 5
    Certificate Status Type: OCSP (1)
    Responder ID list Length: 0
    Request Extensions Length: 0
```

When the client includes the status_request extension in the ClientHello, it signals to the server that it is interested in receiving the OCSP (Online Certificate Status Protocol) response (OCSP Stamping) along with the server's certificate.

```
▾ Extension: supported_versions (len=5)
    Type: supported_versions (43)
    Length: 5
    Supported Versions length: 4
    Supported Version: TLS 1.3 (0x0304)
    Supported Version: TLS 1.2 (0x0303)
```

The supported_versions extension in the ClientHello provides the server with information about the TLS versions that the client can use. The server then selects the highest mutually supported version for the connection.

```
▸ Extension: psk_key_exchange_modes (len=2)
    Type: psk_key_exchange_modes (45)
    Length: 2
    PSK Key Exchange Modes Length: 1
    PSK Key Exchange Mode: PSK with (EC)DHE key establishment (psk_dhe_ke) (1)
```

The psk_key_exchange_modes extension indicates the key exchange modes supported by the client.

9. Does ClientHello Record contain the Signature_algorithms extension? What's its purpose?
   Ans:

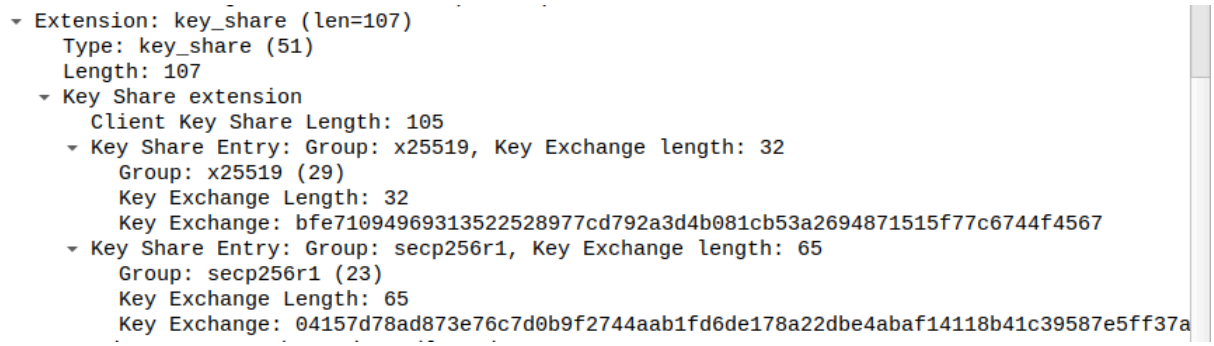
```
▾ Extension: signature_algorithms (len=24)
    Type: signature_algorithms (13)
    Length: 24
    Signature Hash Algorithms Length: 22
  ▾ Signature Hash Algorithms (11 algorithms)
    ▸ Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
    ▸ Signature Algorithm: ecdsa_secp384r1_sha384 (0x0503)
    ▸ Signature Algorithm: ecdsa_secp521r1_sha512 (0x0603)
    ▸ Signature Algorithm: rsa_pss_rsae_sha256 (0x0804)
    ▸ Signature Algorithm: rsa_pss_rsae_sha384 (0x0805)
    ▸ Signature Algorithm: rsa_pss_rsae_sha512 (0x0806)
    ▸ Signature Algorithm: rsa_pkcs1_sha256 (0x0401)
    ▸ Signature Algorithm: rsa_pkcs1_sha384 (0x0501)
    ▸ Signature Algorithm: rsa_pkcs1_sha512 (0x0601)
    ▸ Signature Algorithm: ecdsa_sha1 (0x0203)
    ▸ Signature Algorithm: rsa_pkcs1_sha1 (0x0201)
```

   The Signature_algorithms extension is crucial for ensuring that the client and server can agree on a common set of cryptographic algorithms for securing the handshake and validating digital signatures. That's the only way both can authenticate each other's certificate.

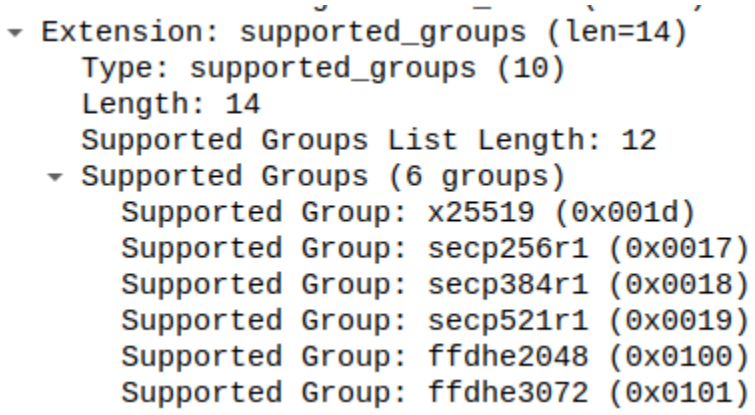
10. Does the client offer any Random number, key share, Supported Groups and PSK in ClientHello Record? How will be these used by the Server?
    Ans:

```
▾ Random: 220c60d2ee3c8ad3e6dba797d8ed14ae77c765077cd7dbccf74d6845253651b7
    GMT Unix Time: Feb  7, 1988 18:19:22.000000000 IST
    Random Bytes: ee3c8ad3e6dba797d8ed14ae77c765077cd7dbccf74d6845253651b7
```

   Random Number at the client's end ensures freshness of the session by preventing any attacker from fooling the client by impersonating a server. It basically guards against replay attacks. The server uses it as one of the parameters towards key material generation.



The Key Share extension is used to convey the client's supported key exchange groups and its public key share for those groups. The client offers its public key share for the selected key exchange groups, allowing the server to use this information during the key exchange process



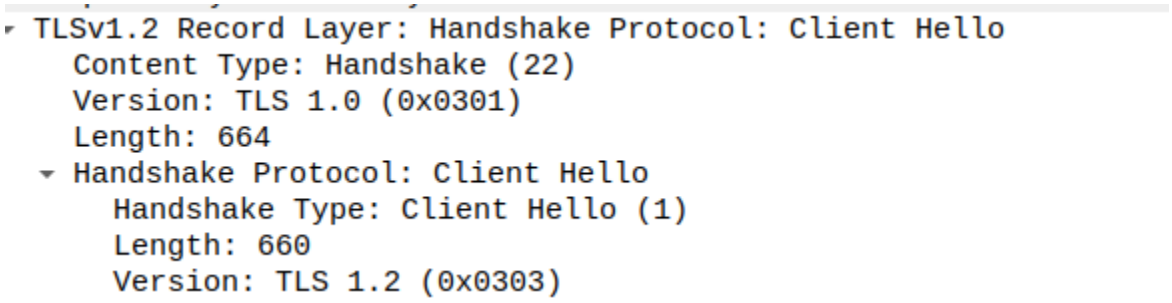
The Supported Groups extension indicates the elliptic curve groups supported by the client. It informs the server about the elliptic curves supported by the client for key exchange. The server can then choose a compatible curve for the key exchange.

It does not not contain PKS as it's the first time the client communicates with the server it will not have a PKS. But the purpose of PKS is to allow session resumption.

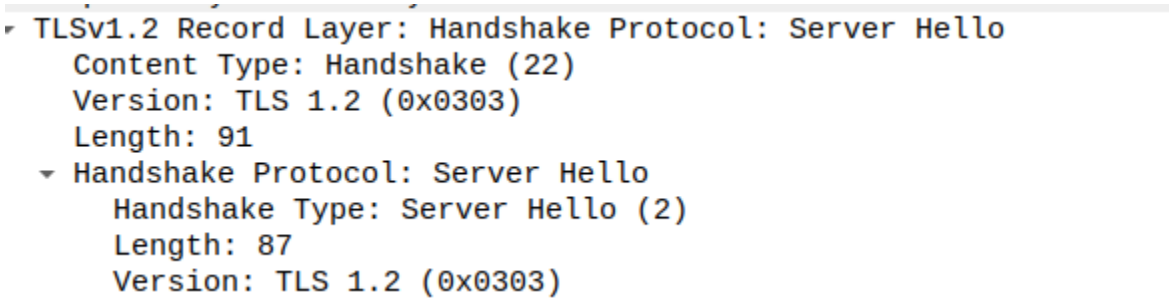
11. What TLS versions your browser/client is supporting? Which one the server selected? Is it the same value as that used in the Record layer header and the Handshake header? Explain.
Ans: My browser/ Client is supporting TLS 1.2 and 1.3 as mentioned in the support version extensions of client hello. The server selected version 1.2 .

At the client side the record layer header and handshake header are different whereas they are same in case of server.

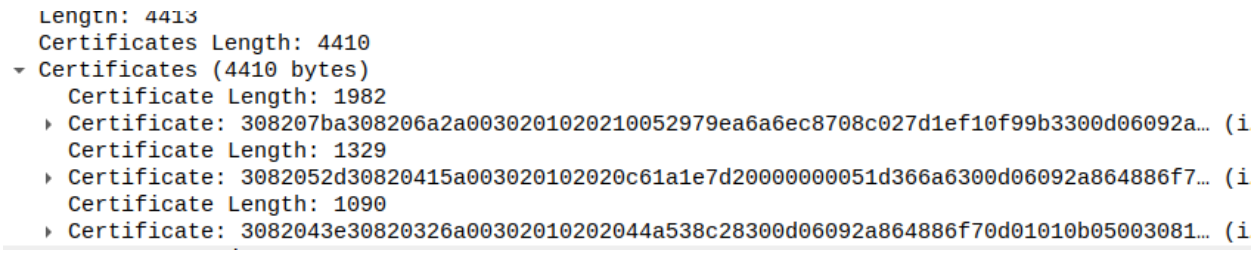
Client's side



Server's side



12. Look at Certificate Record from the server to the client: How many certificates did the server return and how are they related? Who is the issuer of the Bank's certificate? What type of public key the bank is using?
Ans: The server returned three certificates. Each of the certificates in the chain is signed by the previous one. The Certificate Length field in the TLS handshake message relates the certificates.



Issuer: id-at-commonName=Entrust Certification Authority - L1M,
id-at-organizationalUnitName=(c) 2014 Entrust, Inc.
Public Key: algorithm (rsaEncryption)

```
subjectPublicKey: 3082010a0282010100c8cef7bc8e91db52c049d587aab22e4e04aca1b7a3888d18c85d89…
    modulus: 0x00c8cef7bc8e91db52c049d587aab22e4e04aca1b7a3888d18c85d89743fa87c8bd2bab8…
    publicExponent: 65537
```

13. Comment on the key exchange algorithm agreed upon, what are the parameters that got exchanged between client and server to derive the session keys.
    Ans: Among all the cipher suites that the client exchanged with the server selected the below mentioned suit.

    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)

    The key exchange algorithm agreed is ECDHE.

    The following things are exchanged between the client and the server to generate the session key:

    (1) Key share from the client
    (2) Client random
    (3) Server random
    (4) Server Key Exchange
    (5) Client key Exchange

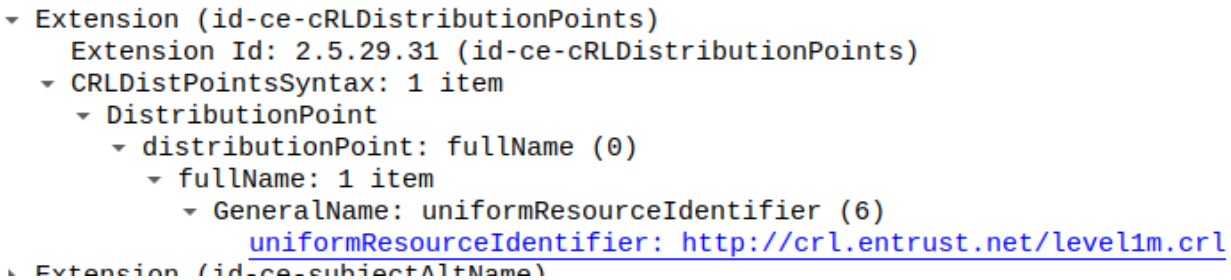
14. Which certificate type (DV/OV/EV) the bank is using?
    Ans: The inclusion of organizational details and OID in certificate policies indicates that the certificate validation level is Extended Validation

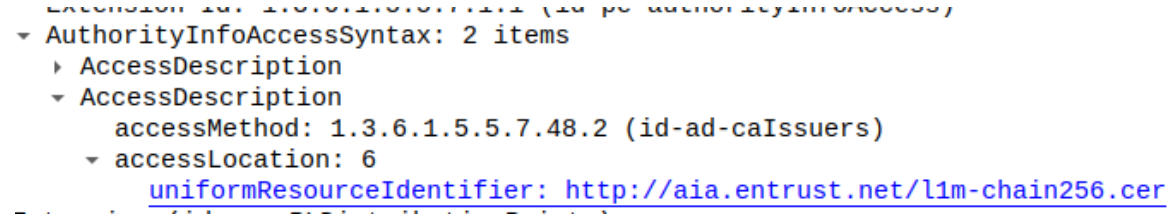
```
RDNSequence item: 1 item (id-at-countryName=US)
RDNSequence item: 1 item (id-at-organizationName=Entrust, Inc.)
RDNSequence item: 1 item (id-at-organizationalUnitName=See www.entrust.net/legal-terms)
RDNSequence item: 1 item (id-at-organizationalUnitName=(c) 2014 Entrust, Inc. - for authorized use only)
RDNSequence item: 1 item (id-at-commonName=Entrust Certification Authority - L1M)
Extension (id-ce-certificatePolicies)
    Extension Id: 2.5.29.32 (id-ce-certificatePolicies)
  CertificatePoliciesSyntax: 2 items
```

15. Which certificate type (single or multi-domain or wild-card) the bank is using?
    Ans: The website uses multi-domain certificate type

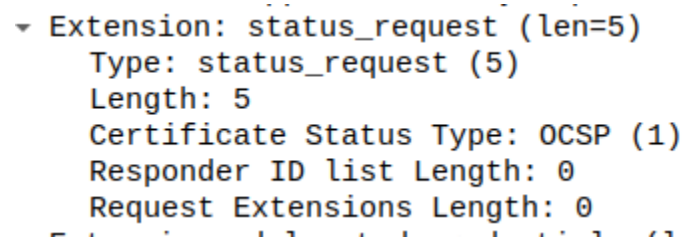
```
Extension (id-ce-subjectAltName)
    Extension Id: 2.5.29.17 (id-ce-subjectAltName)
  GeneralNames: 6 items
    GeneralName: dNSName (2)
        dNSName: www.bankofamerica.com
    GeneralName: dNSName (2)
        dNSName: mobile.bankofamerica.com
    GeneralName: dNSName (2)
        dNSName: smallbusinessonlinecommunity.bankofamerica.com
    GeneralName: dNSName (2)
        dNSName: chatui.ml.com
    GeneralName: dNSName (2)
        dNSName: chatui.merrill.com
```

16. How can the client check whether the certificate is revoked or not: OCSP/CRL? Do the client and server support OCSP stapling?
Ans: Grabbing the list or status in the below mentioned URL's the client can check whether the certificate is received or not.



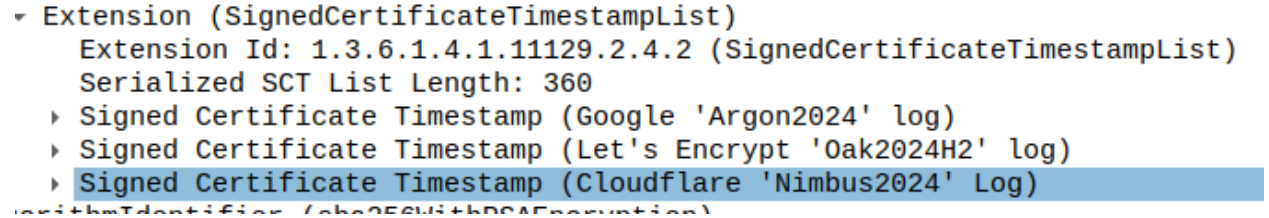


Client supports the OCSP stapling



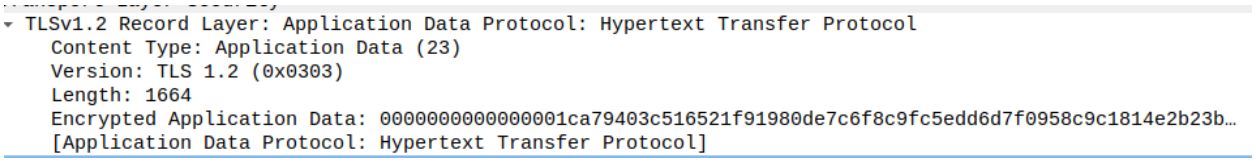
Server might support but doesn't express OCSP stapling status alongside the certificate.

17. How many log servers logged the certificate of the bank? What role does the log server play in the Web PKI ecosystem? Refer: SCT extension.\
Ans: Certificate Transparency is a mechanism designed to enhance the security of the SSL/TLS certificate system by providing a publicly auditable log of all issued certificates. A total of three log server logged certificates of the bank.



18. How is the application data being encrypted? Do the records containing application data include a separate MAC? Does Wireshark distinguish between the encrypted application data and the MAC?

    Ans: In TLS (Transport Layer Security), the application data is encrypted using symmetric key encryption algorithms, typically negotiated during the TLS handshake. Yes, the record containing the application data includes a separate MAC. The encryption process involves two main components: the encryption algorithm (which provides confidentiality) and a MAC (Message Authentication Code) or an AEAD (Authenticated Encryption with Associated Data) tag (which provides integrity and authenticity).Wireshark doesn't explicitly distinguish between the encrypted data and the MAC/AEAD tag in the decrypted view.



19. Look at various keys logged in the file pointed to by the SSLKEYLOGFILE environment variable in your host OS and describe their usage. Also comment on how they are derived from nonces and other parameters using HKDF. Which entity in your system does this job on-the-fly?

    Ans:

    The keys typically logged include:

    Client Write Key (client_write_key):
    Usage: Used by the client to encrypt application data that is sent to the server.
    Derivation: Derived from the negotiated pre-master secret during the TLS handshake using the HKDF (HMAC-based Extract-and-Expand Key Derivation Function) algorithm.

    Server Write Key (server_write_key):
    Usage: Used by the server to encrypt application data that is sent to the client.
    Derived from the negotiated pre-master secret during the TLS handshake using HKDF.

    Client MAC Key (client_write_MAC_key):
    Usage: Used by the client to compute the MAC (Message Authentication Code) for integrity and authenticity verification of outgoing data.
    Derivation: Derived from the client write key using HKDF.

Server MAC Key (server_write_MAC_key):
Usage: Used by the server to compute the MAC for integrity and authenticity verification of outgoing data.
Derivation: Derived from the server write key using HKDF.

Client IV (client_write_IV):
Usage: Initialization Vector (IV) used in the encryption process to ensure that the same plaintext does not encrypt to the same ciphertext.
Derivation: Derived from the client write key using HKDF.

Server IV (server_write_IV):
Usage: Initialization Vector (IV) used by the server in the encryption process.
Derivation: Derived from the server write key using HKDF.

On-the-fly Derivation: The entity responsible for deriving keys on-the-fly during the TLS handshake is usually the TLS implementation library used by the client and server, in my case its NSS

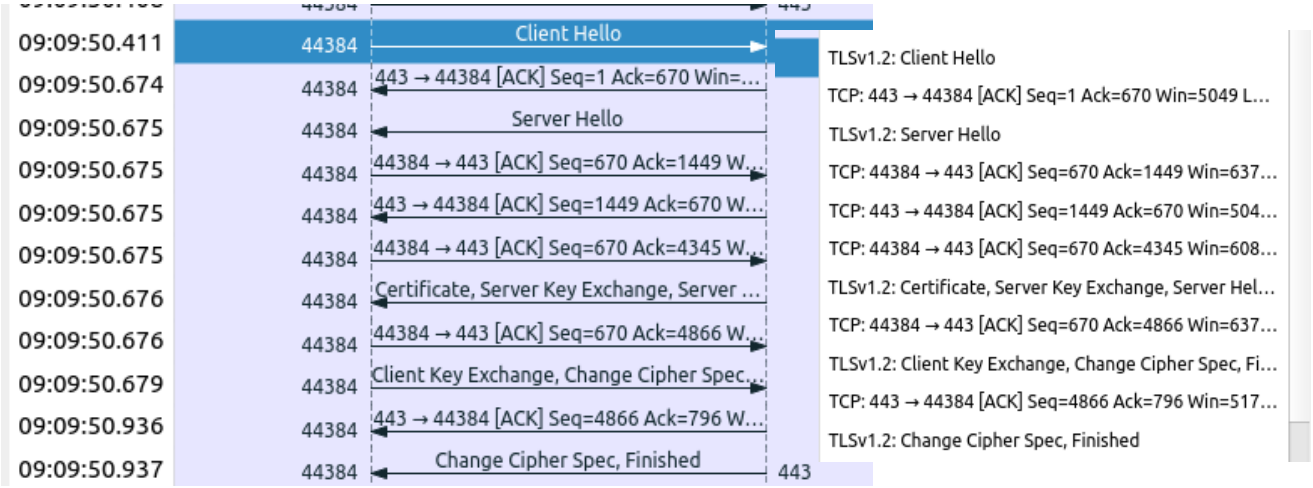
20. Do you see any support for session resumption in the trace? What do you find inside the session ticket, if it is used? Is it based on Session ID/Session ticket or PSK based Session ticket? Do the session IDs play any role in TLS 1.3?

    Ans: The client showed its interest in session resumption by sending zero session ID to the server. Session tickets typically contain timestamp (for validity), cipher suites and Master Secret all encrypted using STEK (session ticket encryption key) and MAC for integrity check. It's based on session ID. In TLS 1.3 Session IDs are still present, but their purpose has shifted. They are used as a legacy mechanism for supporting session resumption in conjunction with Session Tickets

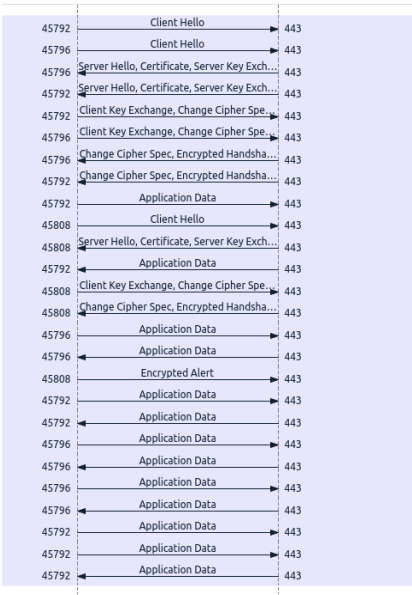
21. How long does it take for TLS to establish a secure (TLS) pipe? How much of it could be reduced when session resumption is used? You may have to revisit the bank site after a while to force session resumption. Answer this question by looking at the flow graph feature in wireshark.

    Ans: As shown in the below image the client sends client hello at time stamp **09:09:50.411** and the last handshake message i.e the client finished got exchanged at timestamp **09:09:50.679**. So the total time it took to establish a TLS pipe is **238 ms.**

1678	09:09:50.411	172.19.124.246	171.161.100.100	TLSv1.2	737	www.bankofamerica.com	Client Hello
1679	09:09:50.650	171.161.100.100	172.19.124.246	TCP	72		443 → 44390 [SYN, ACK] Seq=0 Ack=1 Win=4380 Len=0 MSS=1436 SACK_PERM TSval=4
1680	09:09:50.650	172.19.124.246	171.161.100.100	TCP	68		44390 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0 TSval=2069667395 TSecr=4009374
1681	09:09:50.651	172.19.124.246	171.161.100.100	TLSv1.2	737	www.bankofamerica.com	Client Hello
1682	09:09:50.674	171.161.100.100	172.19.124.246	TCP	68		443 → 44384 [ACK] Seq=1 Ack=670 Win=5049 Len=0 TSval=4009374119 TSecr=206966
1683	09:09:50.675	171.161.100.100	172.19.124.246	TLSv1.2	1516		Server Hello
1684	09:09:50.675	172.19.124.246	171.161.100.100	TCP	68		44384 → 443 [ACK] Seq=670 Ack=1449 Win=63712 Len=0 TSval=2069667419 TSecr=40
1685	09:09:50.675	171.161.100.100	172.19.124.246	TCP	2964		443 → 44384 [ACK] Seq=1449 Ack=670 Win=5049 Len=2896 TSval=4009374120 TSecr=
1686	09:09:50.675	172.19.124.246	171.161.100.100	TCP	68		44384 → 443 [ACK] Seq=670 Ack=4345 Win=60816 Len=0 TSval=2069667420 TSecr=40
1687	09:09:50.676	171.161.100.100	172.19.124.246	TLSv1.2	589		Certificate, Server Key Exchange, Server Hello Done
1688	09:09:50.676	172.19.124.246	171.161.100.100	TCP	68		44384 → 443 [ACK] Seq=670 Ack=4866 Win=63712 Len=0 TSval=2069667420 TSecr=40
1689	09:09:50.679	172.19.124.246	171.161.100.100	TLSv1.2	194		Client Key Exchange, Change Cipher Spec, Finished



It actually does not resumed the session on revisiting the bank server as shown below:



If the resumption would have happened, it would have reduced the time using partial handshake which does not involve the certificate exchange. Effectively reducing the communication setup time. Everything that exchanges between 09:09:50:675 and 09:09:50:679 might be neglected.

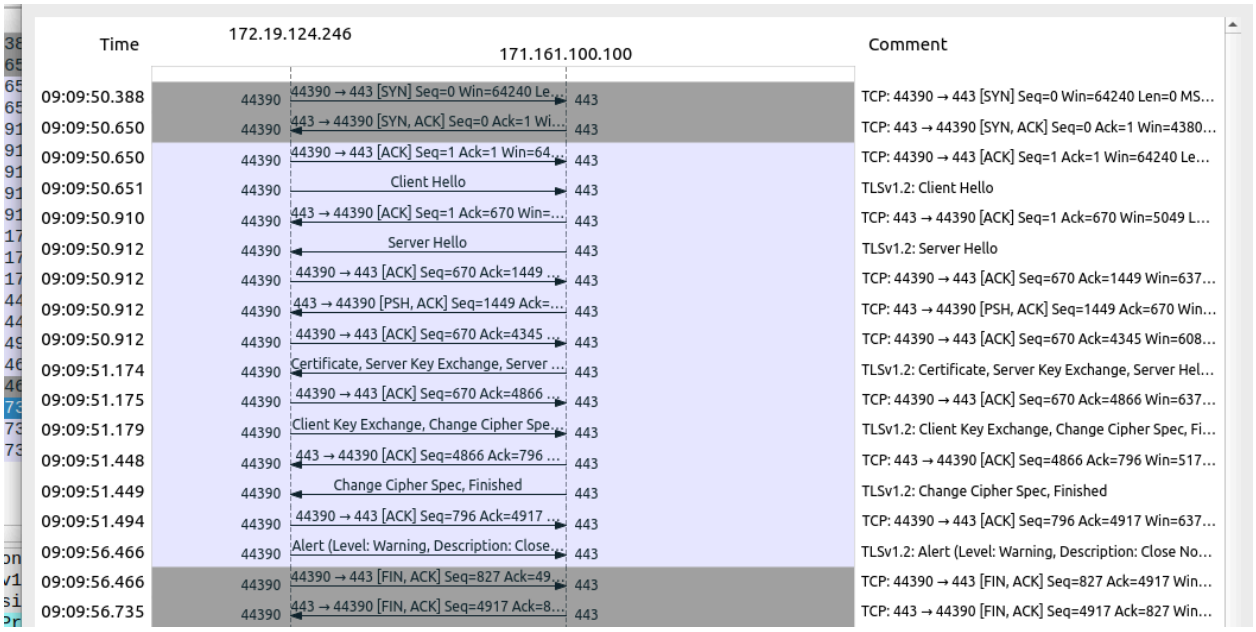
22. What is the duration of the HTTPS session, how many IP packets are exchanged in the browsing session (starting from the first TCP SYN packet till TCP FIN packet)?

Ans: The entire duration of a HTTPS session can be judged based on the time period between the TCP SYN and TCP FIN message. Actually it is somewhere in the time slice after the TLS handshake procedure i.e. after the client finished TCP FIN message. But abstractly it can be what is mentioned earlier.

TCP SYN: 09:09:50:388
TCP FIN: 09:09:56:735
—----------
Total Time: 347 millisecond

Time	Source	Destination	Protocol	Length	Server Name	Info
09:09:50.388	172.19.124.246	171.161.100.100	TCP	76		44390 → 443 [SYN] Seq=0 W
09:09:50.650	171.161.100.100	172.19.124.246	TCP	72		443 → 44390 [SYN, ACK] Se
09:09:50.650	172.19.124.246	171.161.100.100	TCP	68		44390 → 443 [ACK] Seq=1 A
09:09:50.651	172.19.124.246	171.161.100.100	TLSv1.2	737	www.bankofamerica.com	Client Hello
09:09:50.910	171.161.100.100	172.19.124.246	TCP	68		443 → 44390 [ACK] Seq=1 A
09:09:50.912	171.161.100.100	172.19.124.246	TLSv1.2	1516		Server Hello
09:09:50.912	172.19.124.246	171.161.100.100	TCP	68		44390 → 443 [ACK] Seq=670
09:09:50.912	171.161.100.100	172.19.124.246	TCP	2964		443 → 44390 [PSH, ACK] Se
09:09:50.912	172.19.124.246	171.161.100.100	TCP	68		44390 → 443 [ACK] Seq=670
09:09:51.174	171.161.100.100	172.19.124.246	TLSv1.2	589		Certificate, Server Key E
09:09:51.175	172.19.124.246	171.161.100.100	TCP	68		44390 → 443 [ACK] Seq=670
09:09:51.179	172.19.124.246	171.161.100.100	TLSv1.2	194		Client Key Exchange, Chan
09:09:51.448	171.161.100.100	172.19.124.246	TCP	68		443 → 44390 [ACK] Seq=486
09:09:51.449	171.161.100.100	172.19.124.246	TLSv1.2	119		Change Cipher Spec, Finis
09:09:51.494	172.19.124.246	171.161.100.100	TCP	68		44390 → 443 [ACK] Seq=796
09:09:56.466	172.19.124.246	171.161.100.100	TLSv1.2	99		Alert (Level: Warning, De
09:09:56.466	172.19.124.246	171.161.100.100	TCP	68		44390 → 443 [FIN, ACK] Se
09:09:56.735	171.161.100.100	172.19.124.246	TCP	68		443 → 44390 [FIN, ACK] Se

Packet capture of TCP steam follow in wireshark



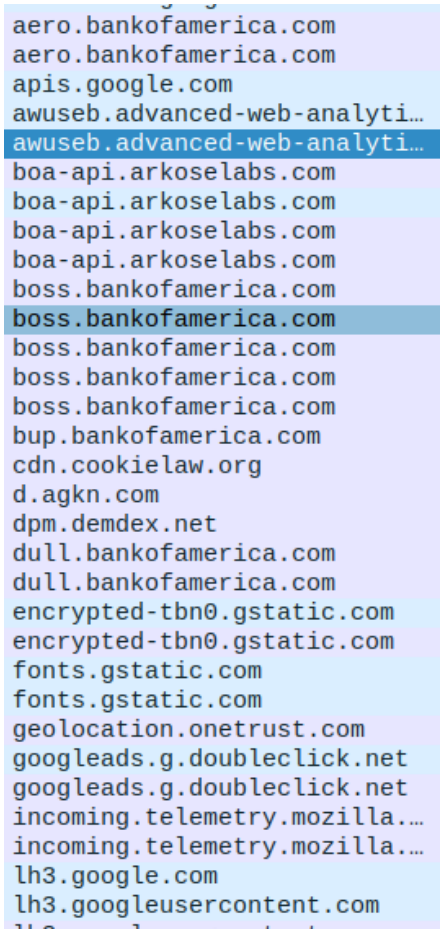
Flow graph of the TCP pipe

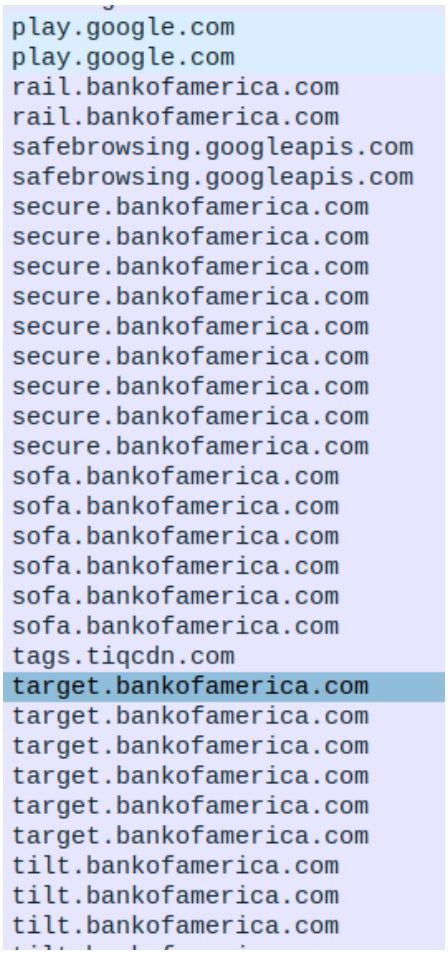
23. How many TLS connections are established with the bank server and its affiliated servers?

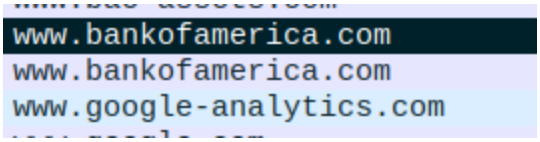
Ans: The Client tried to make 15 requests to the Bank of America server and its affiliated server. The reference of which is attached below. And its name are as follow:

1.	www.bankofamerica.com
2.	aero.bankofamerica.com
3.	boss.bankofamerica.com

4.	bup.bankofamerica.com
5.	dull.bankofamerica.com
6.	rail.bankofamerica.com
7.	secure.bankofamerica.com
8.	sofa.bankofamerica.com
9.	target.bankofamerica.com
10.	tilt.bankofamerica.com
11.	www.google-analytics.com
12.	cdn.cookielaw.org
13.	geolocation.onetrust.com
14.	boa-api.arkoselabs.com
15.	awuseb.advanced-web-analytics.com

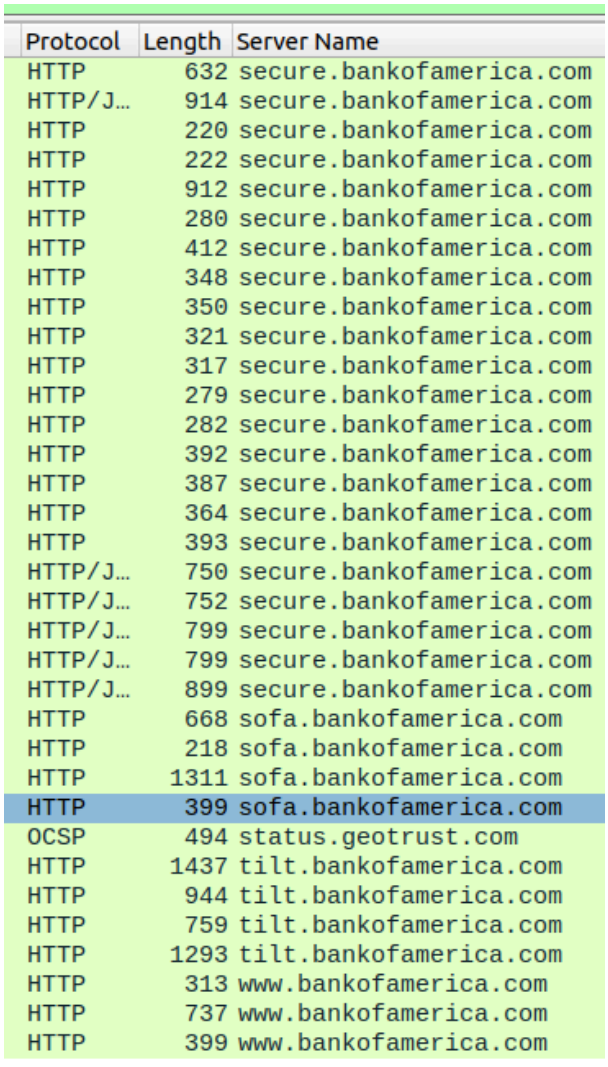






24. How many HTTP request/response packets are exchanged in the browsing session? Identify the packet(s) that carried the response that included the Netbanking LOG-IN page of the bank. Do these response messages carry any security related directives like XSS, sameorigin, HSTS?

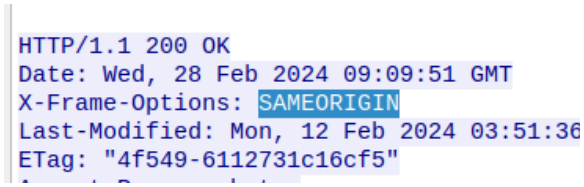
    Ans: Consider the below images:



Protocol	Length	Server Name
HTTP	632	secure.bankofamerica.com
HTTP/J…	914	secure.bankofamerica.com
HTTP	220	secure.bankofamerica.com
HTTP	222	secure.bankofamerica.com
HTTP	912	secure.bankofamerica.com
HTTP	280	secure.bankofamerica.com
HTTP	412	secure.bankofamerica.com
HTTP	348	secure.bankofamerica.com
HTTP	350	secure.bankofamerica.com
HTTP	321	secure.bankofamerica.com
HTTP	317	secure.bankofamerica.com
HTTP	279	secure.bankofamerica.com
HTTP	282	secure.bankofamerica.com
HTTP	392	secure.bankofamerica.com
HTTP	387	secure.bankofamerica.com
HTTP	364	secure.bankofamerica.com
HTTP	393	secure.bankofamerica.com
HTTP/J…	750	secure.bankofamerica.com
HTTP/J…	752	secure.bankofamerica.com
HTTP/J…	799	secure.bankofamerica.com
HTTP/J…	799	secure.bankofamerica.com
HTTP/J…	899	secure.bankofamerica.com
HTTP	668	sofa.bankofamerica.com
HTTP	218	sofa.bankofamerica.com
HTTP	1311	sofa.bankofamerica.com
HTTP	399	sofa.bankofamerica.com
OCSP	494	status.geotrust.com
HTTP	1437	tilt.bankofamerica.com
HTTP	944	tilt.bankofamerica.com
HTTP	759	tilt.bankofamerica.com
HTTP	1293	tilt.bankofamerica.com
HTTP	313	www.bankofamerica.com
HTTP	737	www.bankofamerica.com
HTTP	399	www.bankofamerica.com

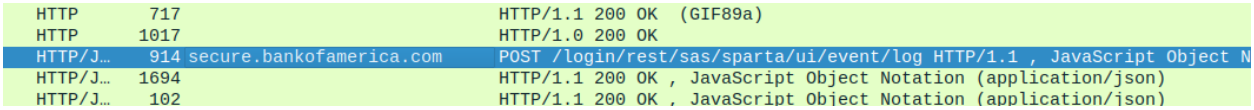
In total there are 6 Http requests/ responses that are exchanged in the browsing session. As in the case of **www.Bankofamerica.com** the website landing page itself contained the login space for net/ internet banking. So the first http request's response will carry the messages for netbanking login details.

The response do contained sameorigin as one of the security directive.

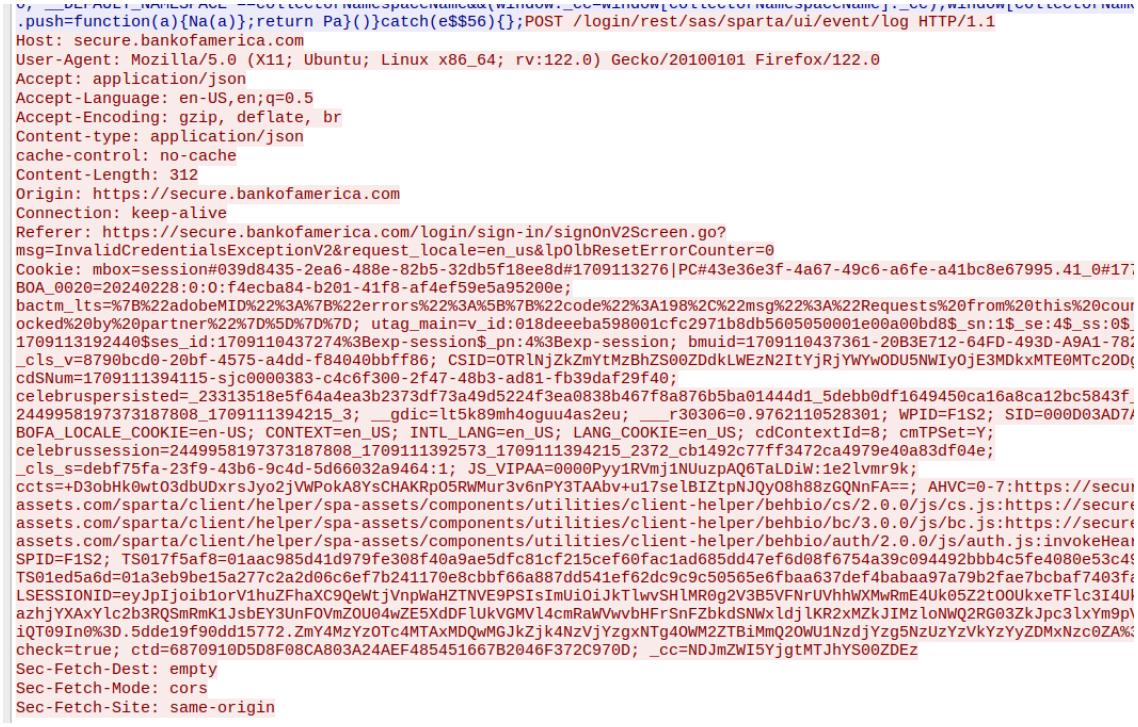


25. Identify the HTTP packet(s) that carried LOG-IN credentials supplied by you. Look at the raw bytes displayed in the wireshark GUI and identify strings that carry your LOG-IN credentials. Are you able to find boh user id and password in the raw packet capture?
    a. It's important that you only keyed in some arbitrary user id and password as part of this assignment for more safety!

Ans: The below entry of the wireshark capture shows the packet which carried Log-in



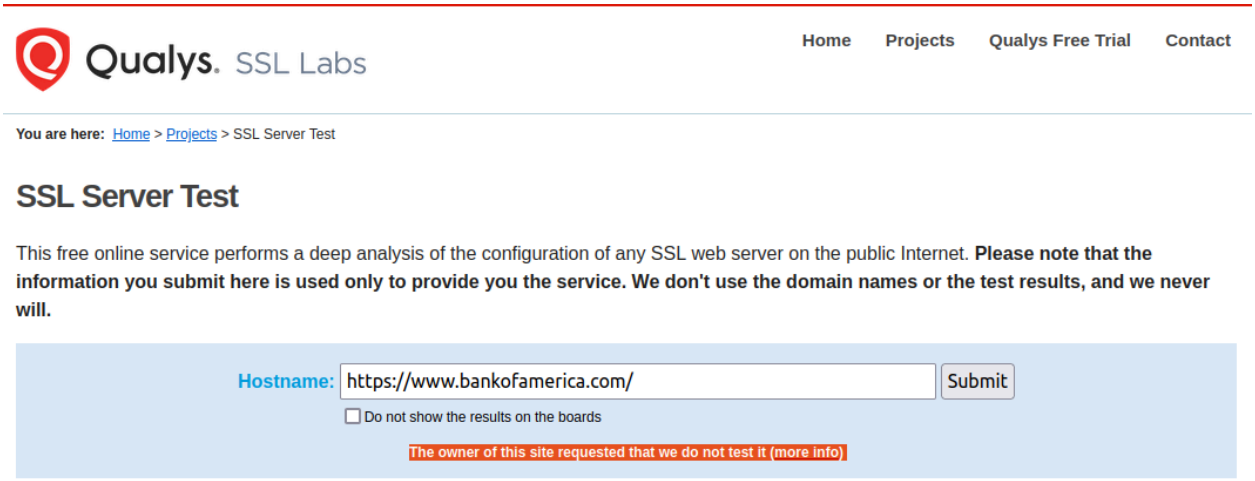
credentials. The login credentials could not be recognized by simply looking at the raw



bytes displayed in the wireshark GUI as they are encrypted and encoded as shown below:

26. Generate an SSL report of the bank using SSL Server Test (Powered by Qualys SSL Labs) and summarize what security features are implemented by the bank's web server for improved online banking by its customers. Does the report flag any issues with the security of the bank?

Ans: **www.bankofamerica.com** has restricted the Qualys SSL Labs from testing and generating SSL reports. The error or warning for the same is attached below:



27. Comment on and explain anything else that you found interesting in the trace!!

    Ans: In the analysis of the captured network traffic, one observation was the client's attempt to establish multiple TLS connections with the server. This behavior could be indicative of parallelization or multiplexing strategies employed by the client, aiming to enhance the efficiency of data retrieval. Multiple TLS connections may be leveraged for concurrent loading of resources, thereby optimizing the overall performance and responsiveness of the web browsing experience.

    Even though the server didn't explicitly announce the use of modern web protocols like HTTP/2 or HTTP/3 in the captured packets, it was interesting to deduce from the traffic that the server is likely employing these advanced protocols. This discovery highlights the server's effort to utilize more efficient and faster ways of communicating on the web, even if the negotiation details aren't explicitly visible in the packet capture.

    It was surprising to find out that even when we visit just one website, our device quietly sets up multiple connections (around 29 TCP according to some research) connections in the background. This happens because of hidden tasks like tracking and fetching resources from services like Google Analytics. It shows how much is going on behind the scenes when we browse the web, even for a single website.

References:

1. Article: K50557518 - Decrypt SSL traffic with the SSLKEYLOGFILE environment variable on Firefox or Google Chrome using Wireshark (f5.com)
2. Wireshark Tutorial: Decrypting HTTPS Traffic (Includes SSL and TLS) (paloaltonetworks.com)
3. Decrypting TLS Streams With Wireshark: Part 1 | Didier Stevens
4. http://www.motobit.com/util/base64-decoder-encoder.asp
5. Dissecting TLS Using Wireshark (catchpoint.com)
6. https://tls13.ulfheim.net/
7. https://www.davidwong.fr/tls13/
8. SSL Server Test (Powered by Qualys SSL Labs)

PLAGIARISM STATEMENT

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarised the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honour violations by other students if I become aware of it.

Name: Yug Patel
Date: 01/03/2024
Signature: Yug Patel