IIT BHUBANESWAR

SYSTEM FORENSIC LAB

ASSIGNMENT-6

NAME - CHANDAN KESHARI
ROLL NUMBER - 24CS06022
BRANCH - CSE(MTECH)

Questions

Wireshark Training

1. List 3 different protocols that appear in the protocol column in the unfiltered packet listing window. What is the MAC address and IP address of your Host?
2. Make a detailed analysis of the packets transmitted and received from your system when you open a web browser and type *https://www.google.com/*
    a. What is the TCP destination port number used for communication and the IP address of the destination?
    b. Is any ARP request ? What is the MAC address of the next hop where packets are communicated?
    c. Is any DNS query made? If yes write the details of each layer information of the packet send and received starting application layer, transport layer , network layer and data links layer?
    d. You can check for a TCP connection being established. Give details of how the three-way handshake is established. Write details about the seq no and window size in the packets getting exchanged.
    e. HTTPS is a secure protocol, which protocol is used for the secure communication. What are the handshake messages being exchanged between the source and destination. Also mention the encryption technique used and the name of the Cipher Suite.
3. Repeat the above process by opening a http site instead of https and write the detail about the analysis made.
    a. Do you find any difference between the http and https protocol. If yes, give the details of the messages exchanged on using http ?
    b. What is the TCP destination port number used for communication and the IP address of the destination?
    c. Which HTTP version is used and the acceptable user agent?
    d. In the HTTP request and response highlight the HTTP header and body. What is the difference between them?
    e. How many HTTP GET request messages were sent by your browser?
    f. How many data-containing TCP segments were needed to carry the single HTTP response?
    g. What is the status code and phrase associated with the response to the HTTP GET request?
    h. Check whether the connection type is persistent or non-persistent HTTPS.

Goal of Assignment:

The objective of this assignment is to provide a comprehensive understanding of network packet analysis using Wireshark. By performing a detailed examination of network traffic, we aim to gain insights into how different protocols operate across the layers of the OSI model. Specifically, this report focuses on:

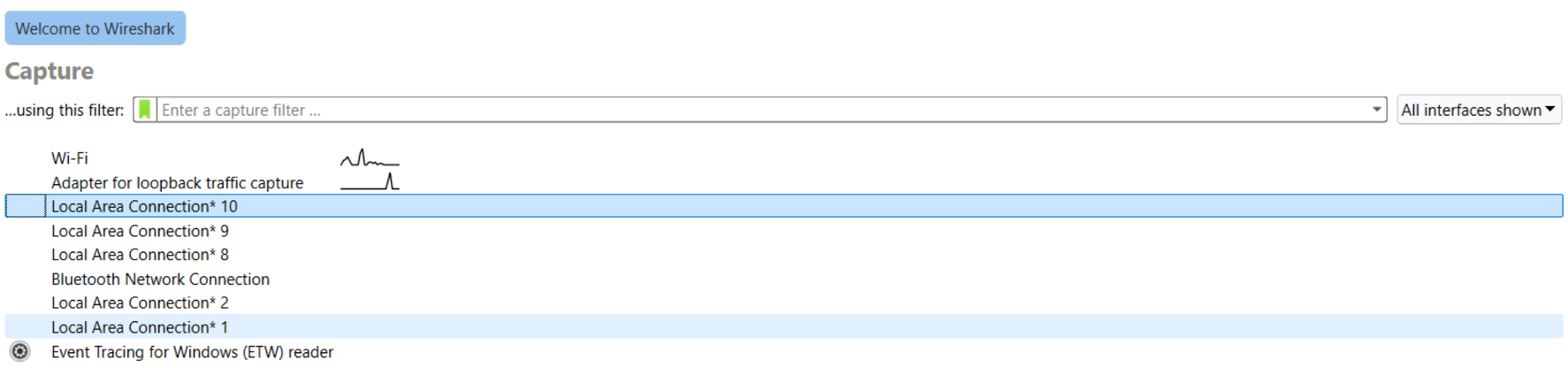
1. **Identification and Analysis of Protocols:** The first step is to observe and list the various protocols that appear in the network traffic, such as TCP, ARP, DNS, HTTP, and HTTPS. This allows us to understand the types of communication taking place in a typical network session.
2. **Analysis of Web Communication (HTTPS):** The second goal is to analyze the network packets transmitted and received when accessing a secure website, such as https://www.google.com/. We will examine key aspects such as:
    - The TCP destination port number and IP address of the destination.
    - ARP requests and the MAC address of the next hop.
    - DNS queries and a layer-by-layer analysis of the packet structure.
    - The TCP three-way handshake, including details about sequence numbers and window sizes.
    - The secure communication protocol used by HTTPS, including handshake messages, encryption techniques, and the name of the Cipher Suite used.
3. **Comparison Between HTTP and HTTPS:** Lastly, we will compare the behavior of non-secure (HTTP) and secure (HTTPS) communication. The report will cover the encryption methods, cipher suites, and structural differences in the messages exchanged during these two types of web communications.

Questions - 1

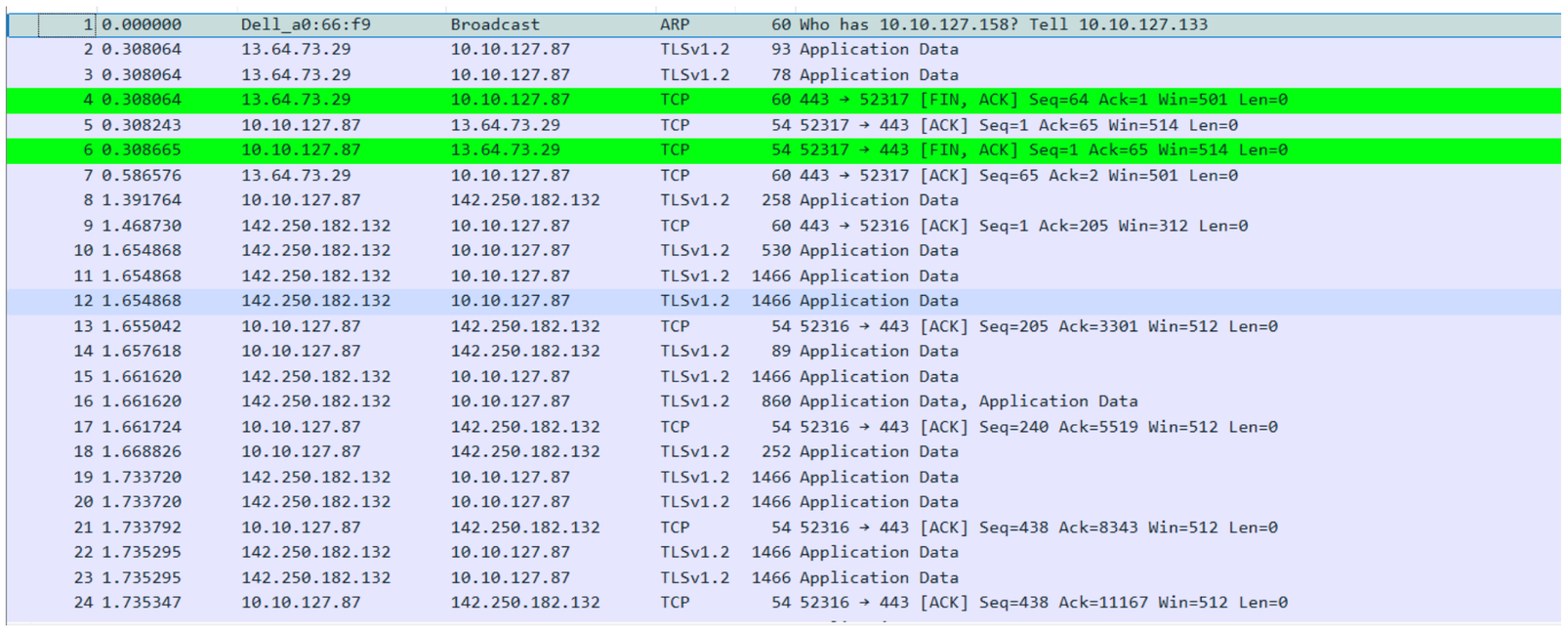
In this first question, we need to tell the any 3 different type of protocols that appear when we capture the network packet without applying any filter & also we need to tell MAC and IP address of our host device. Below is step-by-step procedure to do this task:

Step-1: Capture Network Traffic

First, open Wireshark, then select the one of the packet capture on the network interface that system is using for internet access(e.g. Wi-fi or Ethernet) out of available options.



I have selected the Wi-Fi option as I am using Wi-Fi for internet access. After selecting this, Wireshark will automatically begin capturing packets. To stop capturing of packets, we need to click on red button in top left corner "***Stop capturing packets***". Now we can observe different protocols using in different-2 packets or frames listed in "***Packet list pane***".



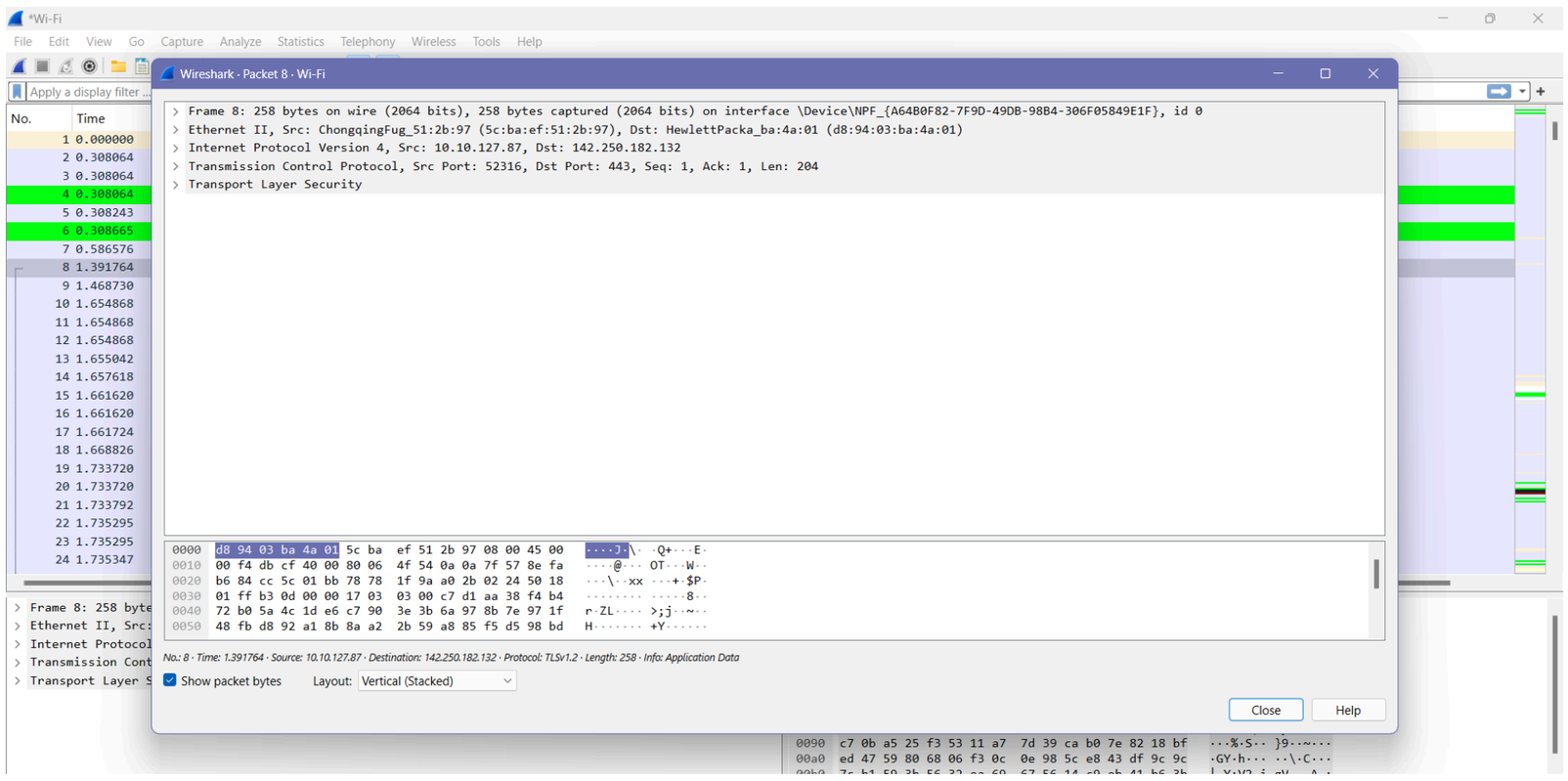
1	0.000000	Dell_a0:66:f9	Broadcast	ARP	60	Who has 10.10.127.158? Tell 10.10.127.133
2	0.308064	13.64.73.29	10.10.127.87	TLSv1.2	93	Application Data
3	0.308064	13.64.73.29	10.10.127.87	TLSv1.2	78	Application Data
4	0.308064	13.64.73.29	10.10.127.87	TCP	60	443 → 52317 [FIN, ACK] Seq=64 Ack=1 Win=501 Len=0
5	0.308243	10.10.127.87	13.64.73.29	TCP	54	52317 → 443 [ACK] Seq=1 Ack=65 Win=514 Len=0
6	0.308665	10.10.127.87	13.64.73.29	TCP	54	52317 → 443 [FIN, ACK] Seq=1 Ack=65 Win=514 Len=0
7	0.586576	13.64.73.29	10.10.127.87	TCP	60	443 → 52317 [ACK] Seq=65 Ack=2 Win=501 Len=0
8	1.391764	10.10.127.87	142.250.182.132	TLSv1.2	258	Application Data
9	1.468730	142.250.182.132	10.10.127.87	TCP	60	443 → 52316 [ACK] Seq=1 Ack=205 Win=312 Len=0
10	1.654868	142.250.182.132	10.10.127.87	TLSv1.2	530	Application Data
11	1.654868	142.250.182.132	10.10.127.87	TLSv1.2	1466	Application Data
12	1.654868	142.250.182.132	10.10.127.87	TLSv1.2	1466	Application Data
13	1.655042	10.10.127.87	142.250.182.132	TCP	54	52316 → 443 [ACK] Seq=205 Ack=3301 Win=512 Len=0
14	1.657618	10.10.127.87	142.250.182.132	TLSv1.2	89	Application Data
15	1.661620	142.250.182.132	10.10.127.87	TLSv1.2	1466	Application Data
16	1.661620	142.250.182.132	10.10.127.87	TLSv1.2	860	Application Data, Application Data
17	1.661724	10.10.127.87	142.250.182.132	TCP	54	52316 → 443 [ACK] Seq=240 Ack=5519 Win=512 Len=0
18	1.668826	10.10.127.87	142.250.182.132	TLSv1.2	252	Application Data
19	1.733720	142.250.182.132	10.10.127.87	TLSv1.2	1466	Application Data
20	1.733720	142.250.182.132	10.10.127.87	TLSv1.2	1466	Application Data
21	1.733792	10.10.127.87	142.250.182.132	TCP	54	52316 → 443 [ACK] Seq=438 Ack=8343 Win=512 Len=0
22	1.735295	142.250.182.132	10.10.127.87	TLSv1.2	1466	Application Data
23	1.735295	142.250.182.132	10.10.127.87	TLSv1.2	1466	Application Data
24	1.735347	10.10.127.87	142.250.182.132	TCP	54	52316 → 443 [ACK] Seq=438 Ack=11167 Win=512 Len=0

From observing "packet list panel", we can see several protocols used in different-2 packets or frame, 3 of them listed below:

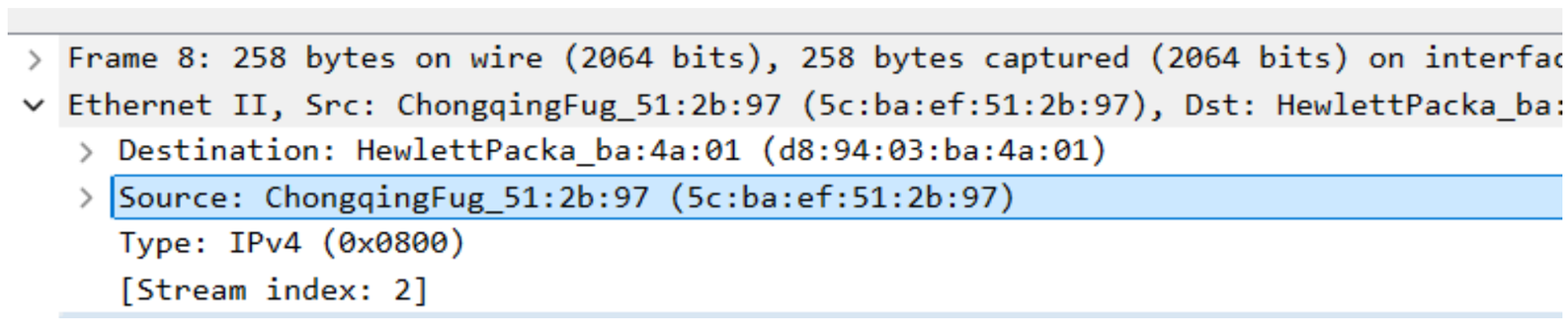
- TCP
- ARP
- TLSv1.2

Step-2: MAC & IP Address of Host

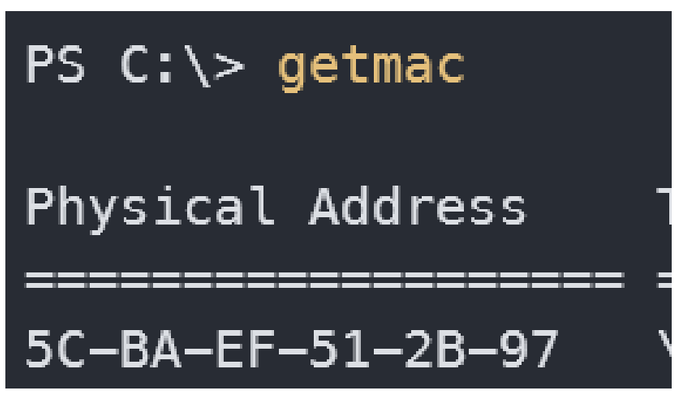
To check MAC & IP address of host system, we need to select a packet from list, then details of the packet will be listed in "***packet detail pane***" usually in bottom left corner or we can double click on packet, then packet detail pane window will pop up which will contain information about packet. Like below:



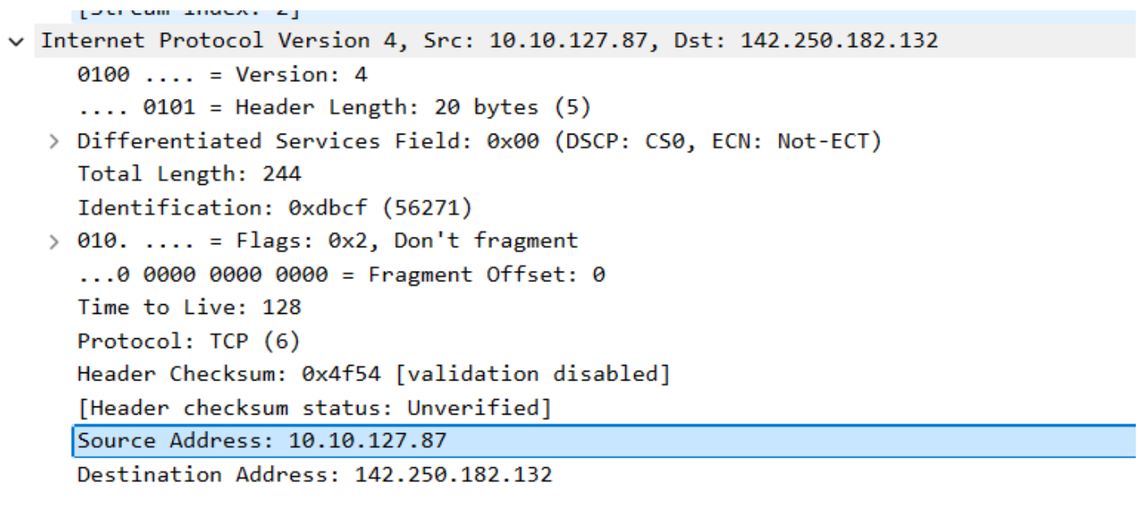
To see MAC address, click on ethernet list, where we get our host MAC address on "***Source***" field. i.e. ***5c : ba : ef : 51 : 2b : 97***. See below:

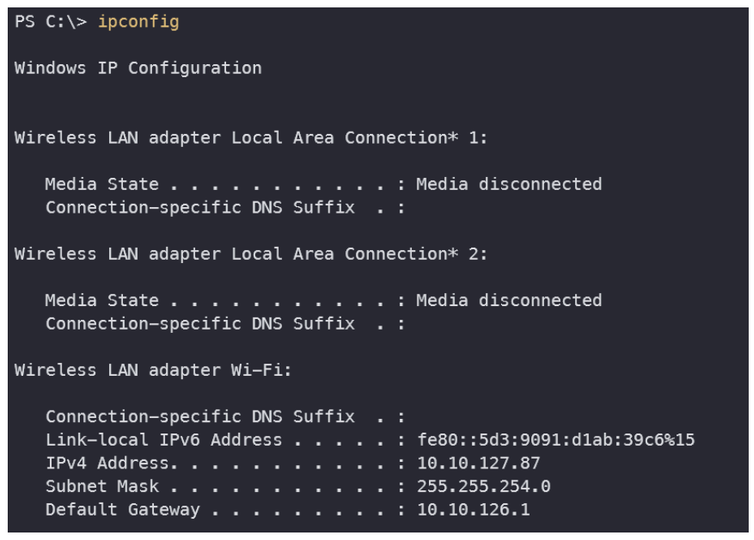


Which we can ensure, by running command "***getmac***" on terminal, see below:



To see IP address, click on "***Internet Protocol***" list, where we get our host IP address on "***Source Address***" field. i.e. ***10.10.127.87***. Which we can also ensure using command "***ipconfig***". See below:



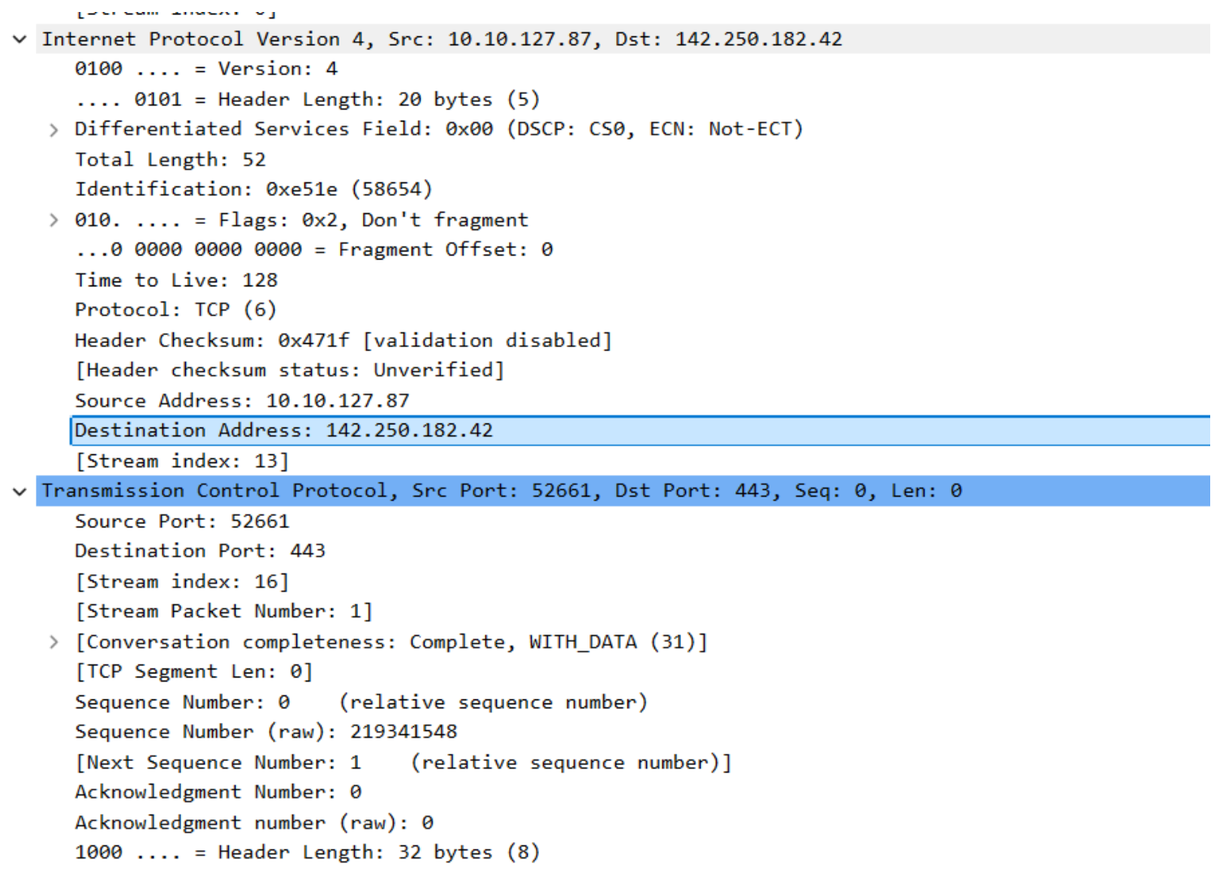


Questions - 2

In this question, we need analysis packets transmitted and received from our system when we open a web browser and type "***https://www.google.com/***".

Part-A

To do this task, we first capture the packet on network traffic, and then filter the HTTPS traffic only by typing "***tcp***" on filter bas. We will sniffed a TCP SYN packet and checked its destination IP address and TCP port number.

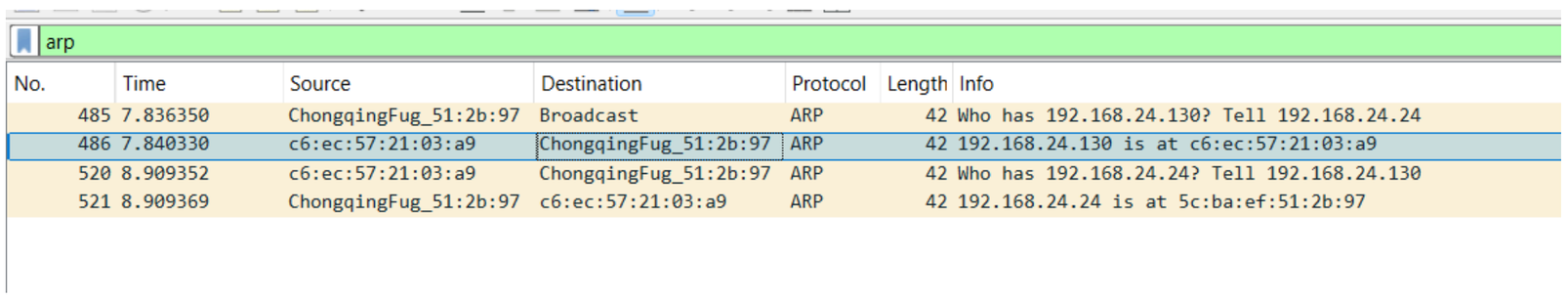


From above diagram, we get IP address & TCP port number for destination:

- IP Address: ***142.250.182.42***
- TCP Port number: ***443***

Part-B

To see that whether we get any ARP request, we can filter packets capture by typing "***arp***" on filter bar, which will display only ARP packets if any.



We can see there are several ARP request packet & the MAC address of the next hop where packets are communicated is:

- MAC Address: ***c6: ec: 57: 21: 03: a9***

```
Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: c6:ec:57:21:03:a9 (c6:ec:57:21:03:a9)
    Sender IP address: 192.168.24.130
    Target MAC address: ChongqingFug_51:2b:97 (5c:ba:ef:51:2b:97)
    Target IP address: 192.168.24.24
```

Part-C

Similarly, To see that whether we get information of each layer of standard DNS query made, we will first filter packets by typing "***dns***" on display filter bar which will list only DNS query made.

dns

No.	Time	Source	Destination	Protocol	Length	Info
43	1.558794	10.10.127.87	8.8.8.8	DNS	88	Standard query 0x5de1 A f-log-extension.grammarly.io
52	1.617091	8.8.8.8	10.10.127.87	DNS	216	Standard query response 0x5de1 A f-log-extension.grammarly.io A 54.197.56.251 A 34.22
145	2.586577	10.10.127.87	8.8.8.8	DNS	74	Standard query 0xe394 A www.google.com
146	2.616121	10.10.127.87	8.8.8.8	DNS	87	Standard query 0x7c72 A encrypted-vtbn0.gstatic.com
147	2.649688	8.8.8.8	10.10.127.87	DNS	90	Standard query response 0xe394 A www.google.com A 142.250.182.132
159	2.723427	10.10.127.87	8.8.4.4	DNS	87	Standard query 0x7c72 A encrypted-vtbn0.gstatic.com
160	2.772885	8.8.8.8	10.10.127.87	DNS	103	Standard query response 0x7c72 A encrypted-vtbn0.gstatic.com A 142.250.195.46
176	2.867792	8.8.4.4	10.10.127.87	DNS	103	Standard query response 0x7c72 A encrypted-vtbn0.gstatic.com A 142.250.195.46
281	4.470595	10.10.127.87	8.8.8.8	DNS	75	Standard query 0x33e7 A www.gstatic.com
282	4.470851	10.10.127.87	8.8.8.8	DNS	77	Standard query 0x4bf0 A fonts.gstatic.com

Information Each Layer of Packet Sent:

First look packet with domain name ***www.google.com,*** then find packet sent from host and then open the packet detail pane to see the information of each layer.

- **Application Layer Information:** Expand "***Domain Name System(DNS)***" to view details about application layer:

```
Domain Name System (query)
    Transaction ID: 0xe394
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  > Queries
    [Response In: 147]
```

- ***Transport Layer Information:*** Expand "***User Datagram Protocol***" to check the transport layer information:

```
User Datagram Protocol, Src Port: 61543, Dst Port: 53
    Source Port: 61543
    Destination Port: 53
    Length: 40
    Checksum: 0x0354 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
    [Stream Packet Number: 3]
  > [Timestamps]
    UDP payload (32 bytes)
```

- **Network Layer Information:** Expand **“Internet Protocol”** to check the network layer information:

```
Internet Protocol Version 4, Src: 10.10.127.87, Dst: 8.8.8.8
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0xbe6d (48749)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0xe2d2 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.10.127.87
    Destination Address: 8.8.8.8
    [Stream index: 5]
```

- ***Data Link Layer Information:*** Expand “***Ethernet***” to find the MAC address information:

```
Ethernet II, Src: ChongqingFug_51:2b:97 (5c:ba:ef:51:2b:97), Dst: HewlettPacka_ba:4a:01 (d8:94:03:ba:4a:01)
  > Destination: HewlettPacka_ba:4a:01 (d8:94:03:ba:4a:01)
  > Source: ChongqingFug_51:2b:97 (5c:ba:ef:51:2b:97)
    Type: IPv4 (0x0800)
    [Stream index: 0]
```

Information Each Layer of Packet Received:

Look for packet with domain name “***www.google.com***” and DNS query with “***response***”. Then open the packet detail pane to see each layer information of received packet:

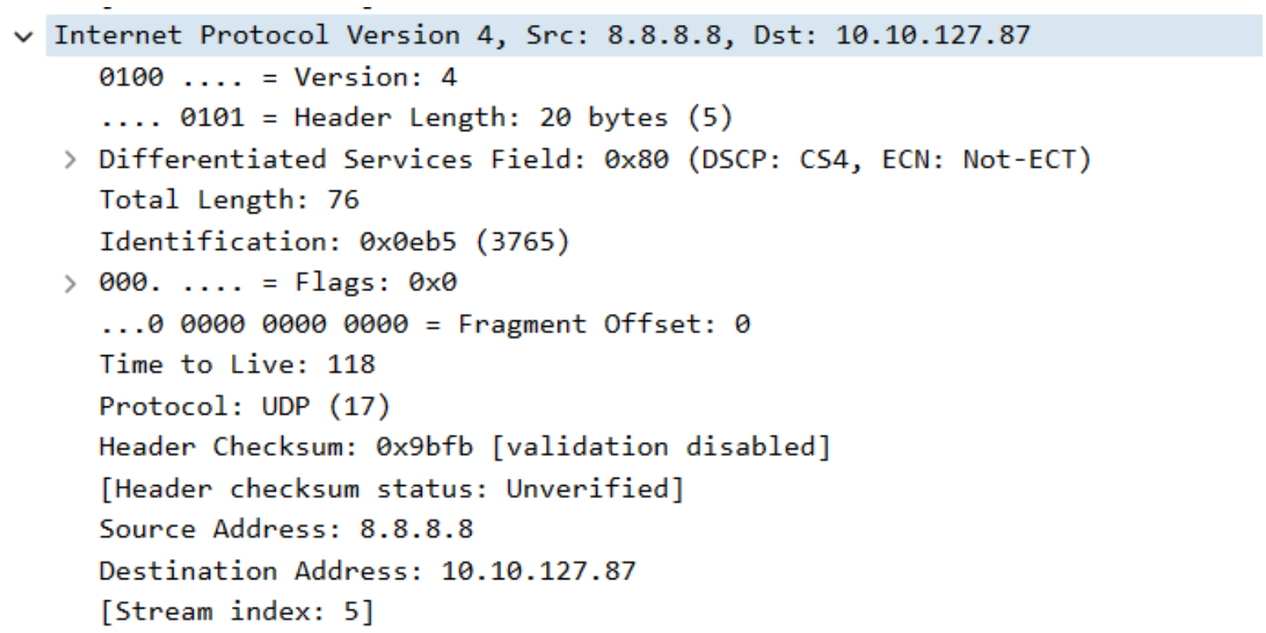
- **Application Layer Information:** Expand **“*Domain Name System(DNS)*”** to view details about application layer:

```
Domain Name System (response)
    Transaction ID: 0xe394
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  > Queries
  > Answers
    [Request In: 145]
    [Time: 0.063111000 seconds]
```

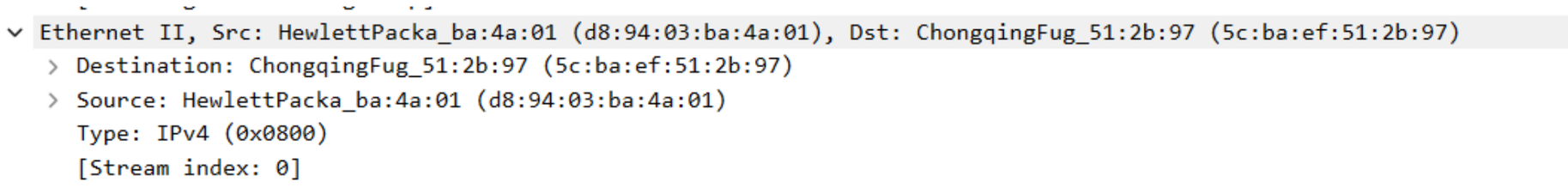
- ***Transport Layer Information:*** Expand “***User Datagram Protocol***” to check the transport layer information:

```
User Datagram Protocol, Src Port: 53, Dst Port: 61543
    Source Port: 53
    Destination Port: 61543
    Length: 56
    Checksum: 0x7cb5 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
    [Stream Packet Number: 4]
  > [Timestamps]
    UDP payload (48 bytes)
```

- **Network Layer Information:** Expand **“Internet Protocol”** to check the network layer information:



- ***Data Link Layer Information:*** Expand “***Ethernet***” to find the MAC address information:



Part-D

Before start doing let’s first see what is ***TCP three-way handshake*** ?

The TCP three-way handshake is a process used to establish a reliable connection between a client and server in a TCP/IP network. Here's how it works:

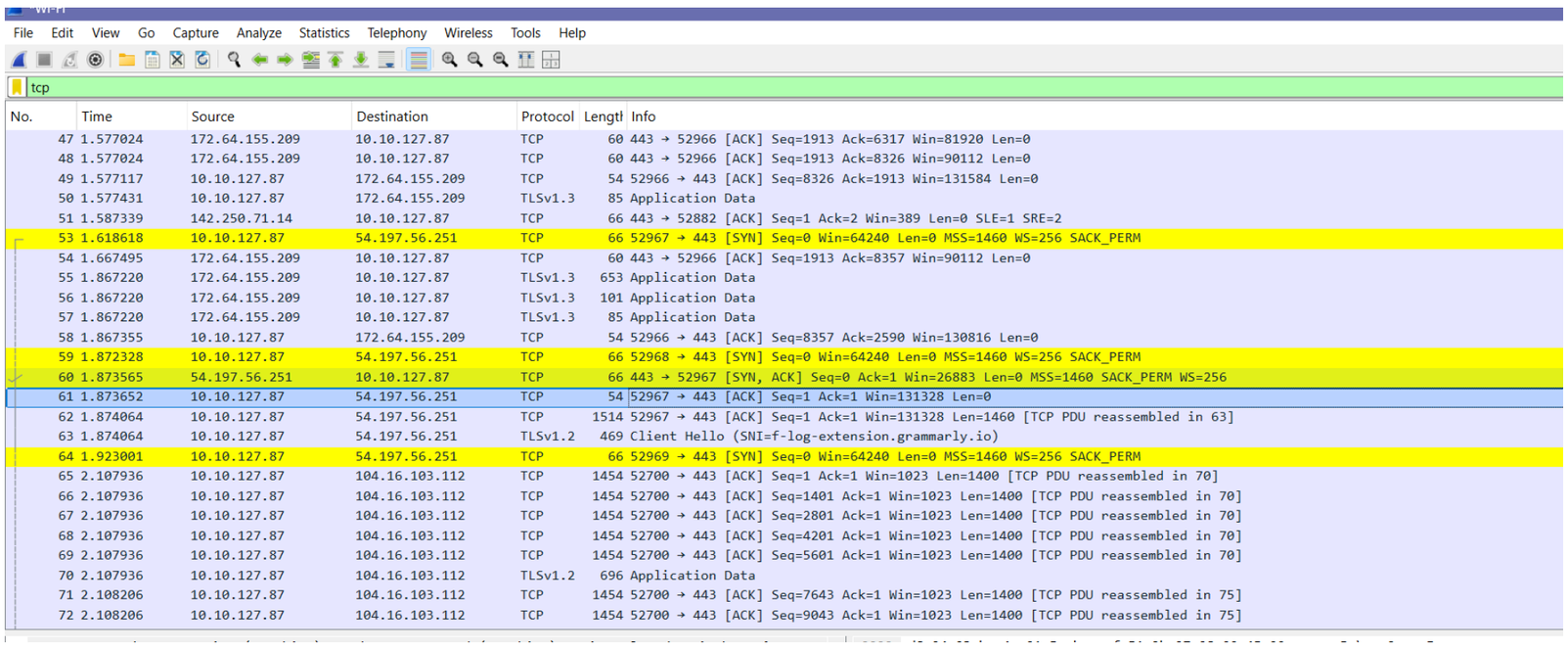
1. ***SYN (Synchronize):*** The client sends a TCP segment with the SYN flag set to initiate the connection. It includes an initial sequence number (Seq).
2. ***SYN-ACK (Synchronize-Acknowledge):*** The server responds with a segment that has both SYN and ACK flags set, acknowledging the client's request. It includes its own sequence number and an acknowledgment number (Ack = client Seq + 1).
3. ***ACK (Acknowledge):*** The client sends an ACK to confirm the connection. This packet acknowledges the server's sequence number (Ack = server Seq + 1).

At the end of this handshake, the connection is established, and both client and server can begin data exchange.

Now, to do this task, we first need to filter packet by typing “***tcp***” which will display only TCP packets. Locate the three-way handshake(three consecutive packets):

- ***SYN:*** Sent by the client to initiate the connection.
- ***SYN-ACK:*** Sent by the server acknowledging the SYN request.
- ***ACK:*** Sent by the client to confirm the connection.

Below, i have attached the screenshot of three-way handshake:



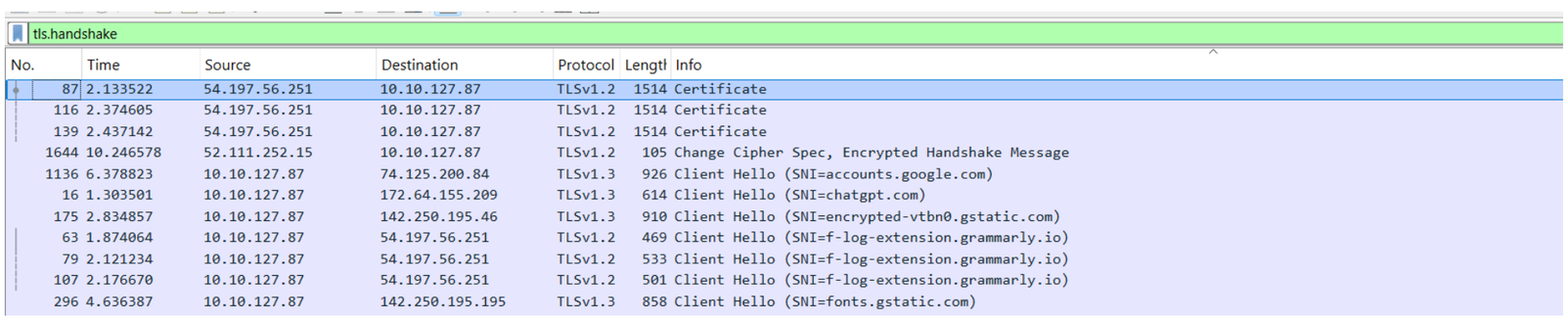
By observing attached screenshot, we can list sequence numbers & window size:

- ***SYN Packet:***
  - Packet No. = 59
  - Seq = 0
  - Window Size = 64240
- ***SYN-ACK Packet:***
  - Packet No. = 60
  - Seq = 0, ACK = 1
  - Window Size = 26883
- ***ACK Packet:***
  - Packet No. = 61
  - Seq = 1, ACK = 1,
  - Window Size = 131328

Part-E

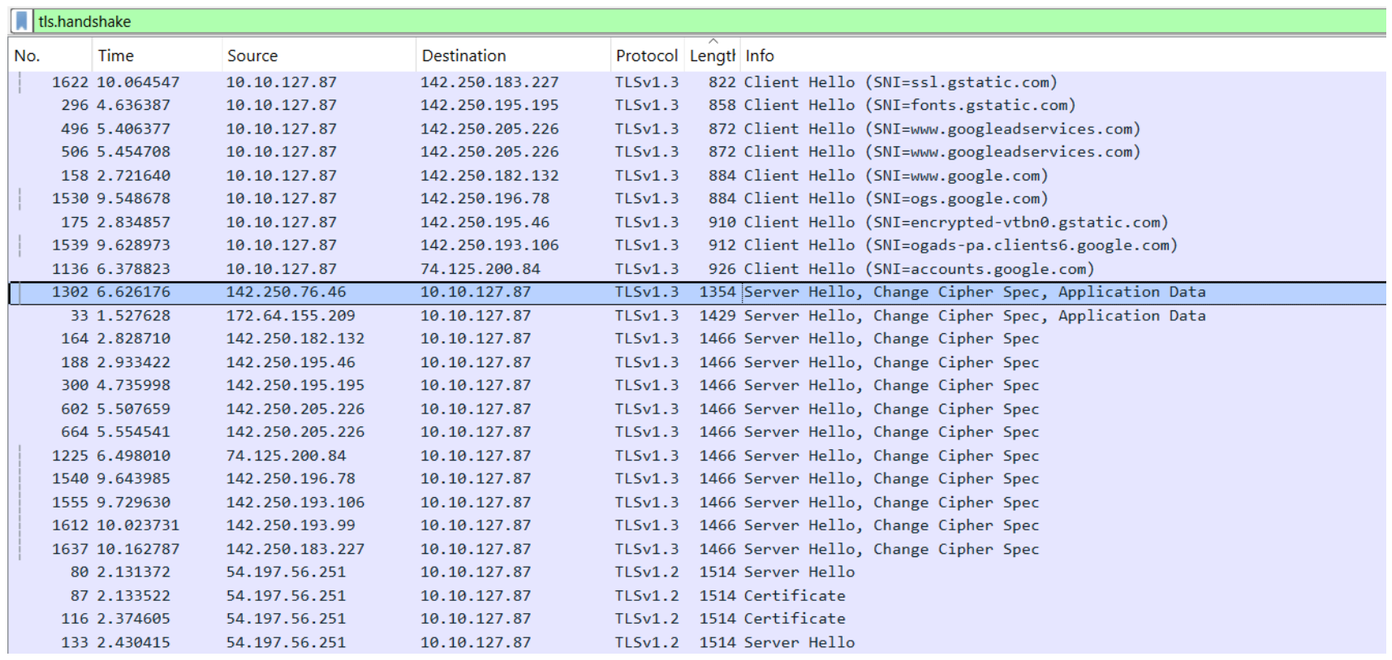
The protocol used for secure communication in HTTPS is TLS(Transport Layer Security) handshake, below is step-by-step procedure:

- **Filter TLS Handshake:** In the filter bar type “tls.handshake” which will display only TLS handshake packets.



- **Locate TLS Handshake Messages:** By observing above screenshot, we can locate below handshake messages:
  - ***Client Hello:*** The client initiates the handshake and proposes a list of supported cipher suites.
  - ***Server Hello:*** The server responds with the selected cipher suite and other configurations.

- ***Certificate:*** The server sends its digital certificate to authenticate itself.
- ***Key Exchange(Optional):*** The server may send key exchange parameters.
- ***Finished:*** Both client and server send this to conclude the handshake and begin secure communication.



- **Cipher Suite:** Click on the ***Client Hello*** or ***Server Hello*** message, in the detail pane, expand the section labeled ***Cipher Suites*** to see the selected encryption method.

From above attached screenshot, we have:

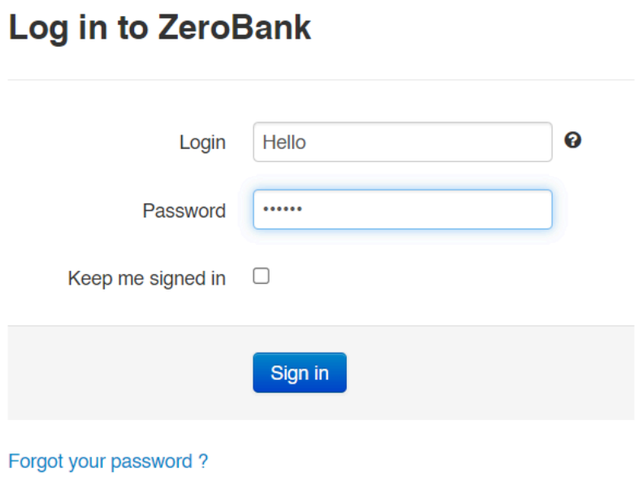
- ***Encryption Technique Used:*** AES (Advanced Encryption Standard)
- ***Cipher Suite Used:*** TLS_AES_128_GCM_SHA256

Questions - 3

In this question, we need analysis the same process as above by opening a HTTP site instead of HTTPS, for this purpose. I have opened:

`http://zero.webappsecurity.com/login.html`

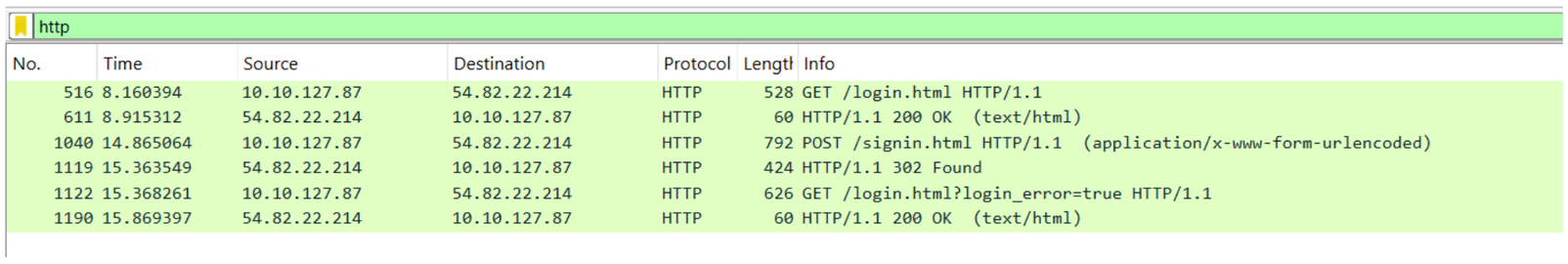
And, typed login = “Hello” and password = “Pillow”:



Now, we capture packets in Wireshark and perform our analysis.

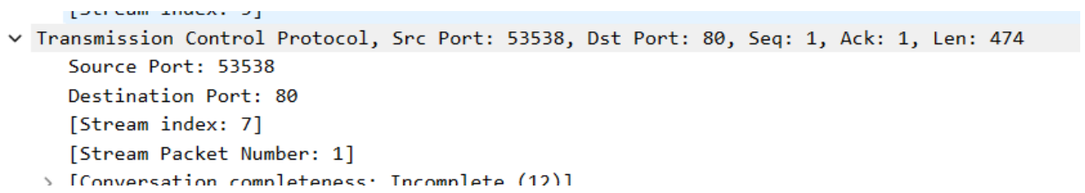
Part-A

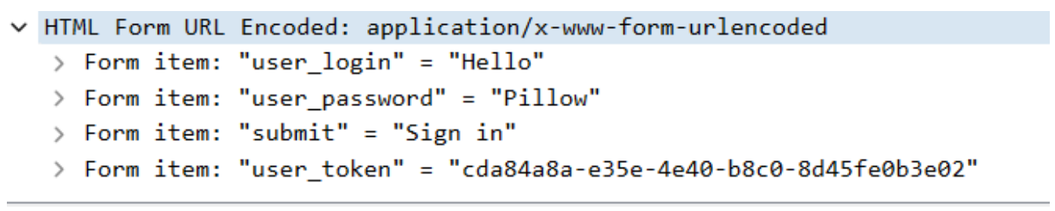
Before answering the question first, let’s filter out HTTP traffic using “***http***” on filter tab which will display only http traffic.



Similarly, we will capture HTTPS traffic and filter packets using “***tls***”, then on observing both the traffic, some difference is found:

- **Encryption:**
  - **HTTP** traffic is unencrypted
  - **HTTPS** is encrypted using TLS
- **Port Number:**
  - **For HTTP:** 80 by default
  - **For HTTPS:** 443 by default
- **Security:**
  - **HTTP:** It sends data in plaintext which we can see in below screenshot
  - **HTTPS:** ensure data CIA tried using encryption and certificates.



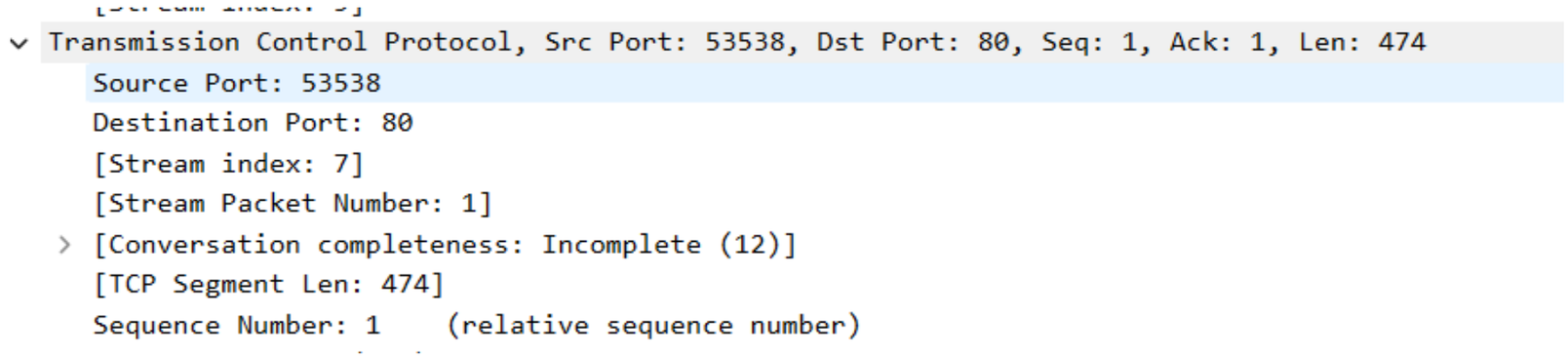


Part-B

To find out the TCP port number and IP address of the destination,

- first filter packets using "**http**" on filter bar,
- then click on any packet to view its details

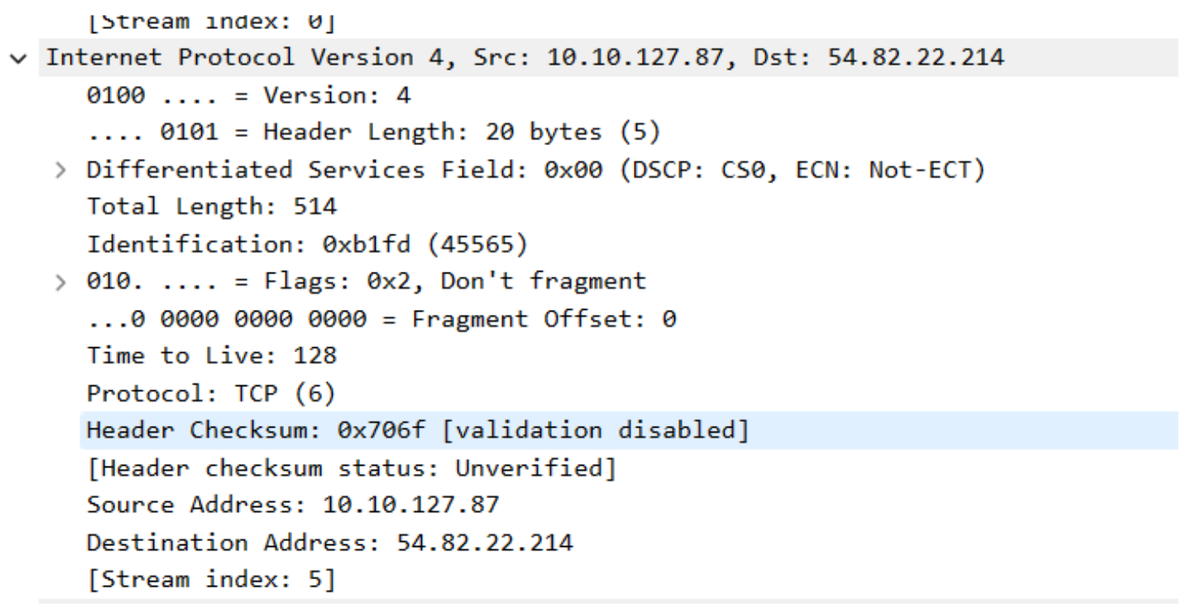
To see TCP detail, in the Packet Details Pane, expand the "Transmission Control Protocol (TCP)" section, where we will find TCP port number of destination:



From above screenshot, we got:

- Destination TCP port number: ***80.***

Similarly, to see IP address of destination, expand "Internet Protocol" section, where we will find our destination IP address:



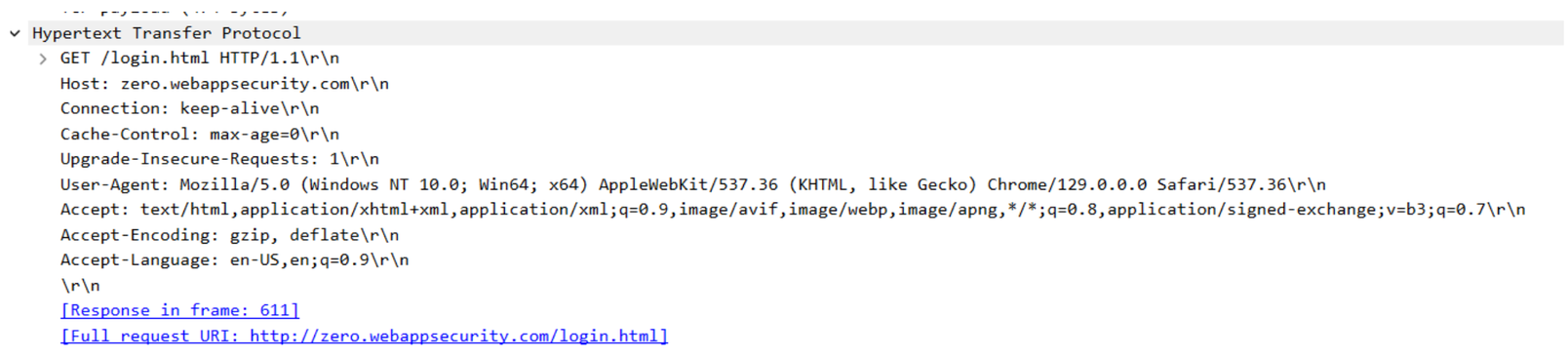
From above screenshot, we got:

- Destination IP Address: ***54.82.22.214***

Part-C

To find out HTTP version used, follow below steps:

- Look for the packet that has either "***GET***" or "***POST***" in the ***Info*** column, click on it.
- In the packet details pane, expand ***Hypertext Transfer Protocol section***.



Here, we get:

- ***HTTP Version:*** HTTP/1.1/r/n
- ***User-Agent:***
  - ***Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/129.0.0.0 Safari/537.36***

Part-D

in this question, we need to list the Header & Body of both request packet and response packet.

Request Packet:

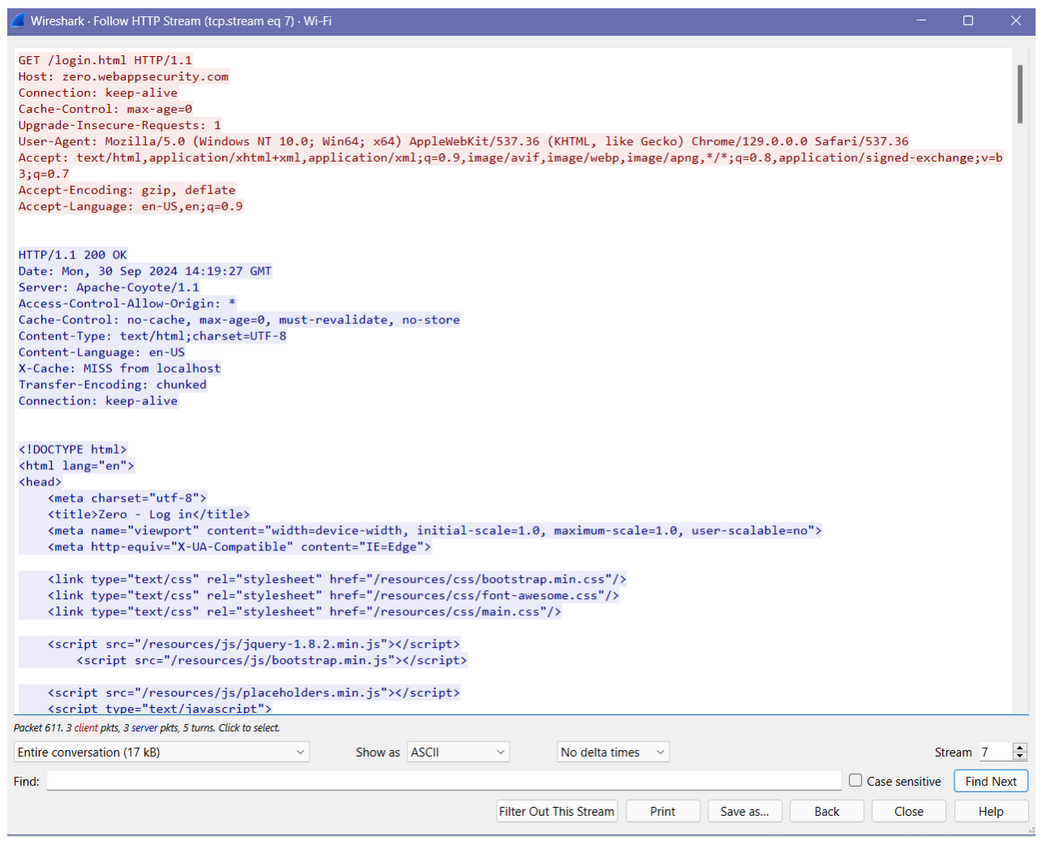
To see the request packet detail, click on packet with "***GET***" in Info column, where we get the HTTP request header, which includes fields like: ***GET***, ***Host***, ***User-Agent***, ***Accept***, etc.

In request packet, we do not have any body part. If request is a POST request, then we may find the body after the header.

Response Packet:

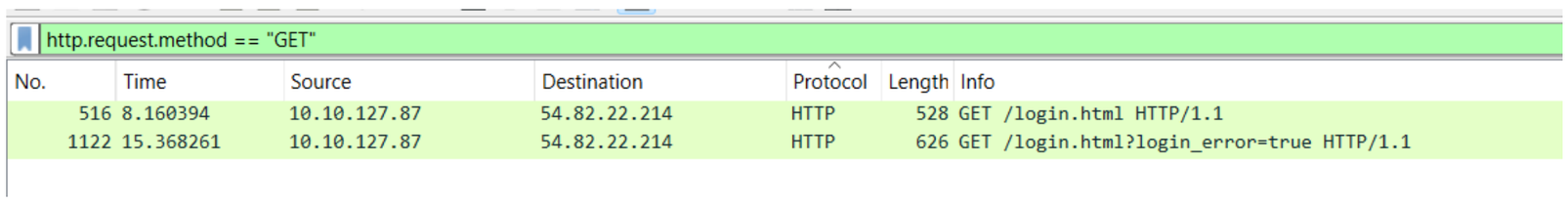
As per request packet header detail, we will find our response frame in ***611*** packet or we can find the response packet that has status code **200 OK** (usually the one following the request packet).
In the packet details pane, expand the Hypertext Transfer Protocol section again.

- ***HTTP Response Header:*** In ***red*** color, look for fields like: *HTTP/1.1 200 OK*, *Content-Type*, *Content-Length*, etc.
- ***HTTP Response Body:*** will be in ***blue*** color



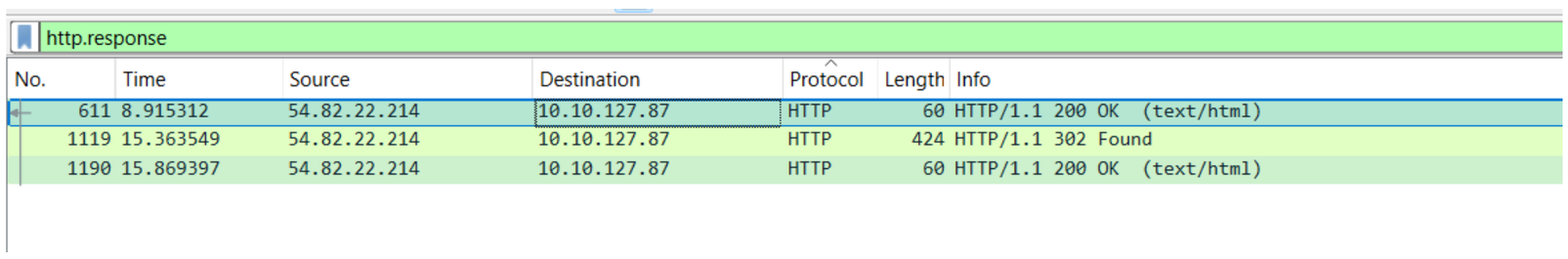
Part-E

To do this task, type "***http.request.method == "GET"***" in filter tab to display only the HTTP GET request packet. Which is 2 request in my case.

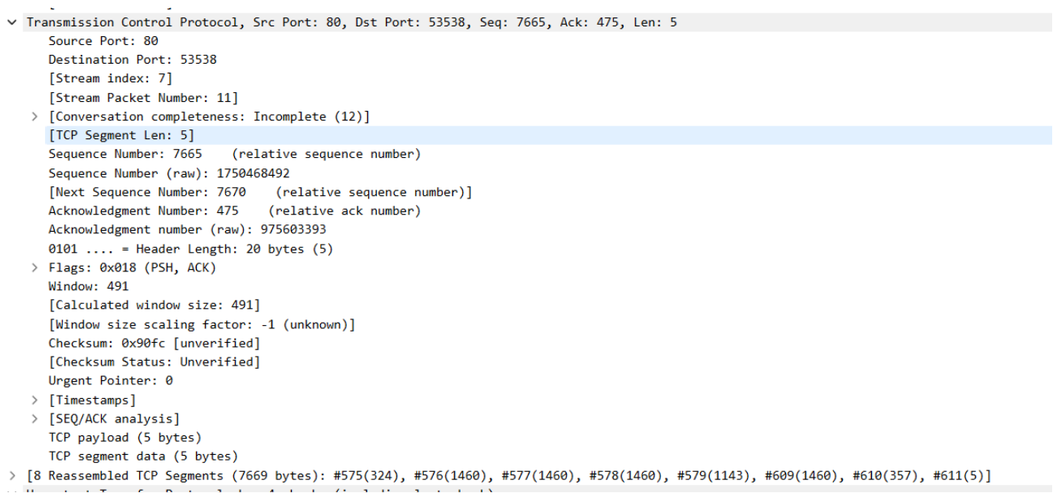


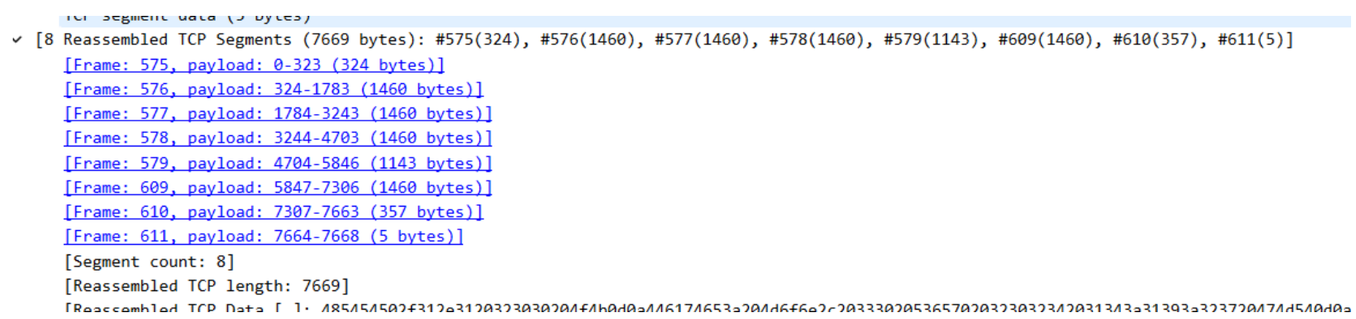
Part-F

First filter the captured packet, using "http.response" in filter tab to display only HTTP response packet:



Now, click on packet with 200 OK response packet, then expand the ***Transmission Control Protocol (TCP)*** section for the selected HTTP response:





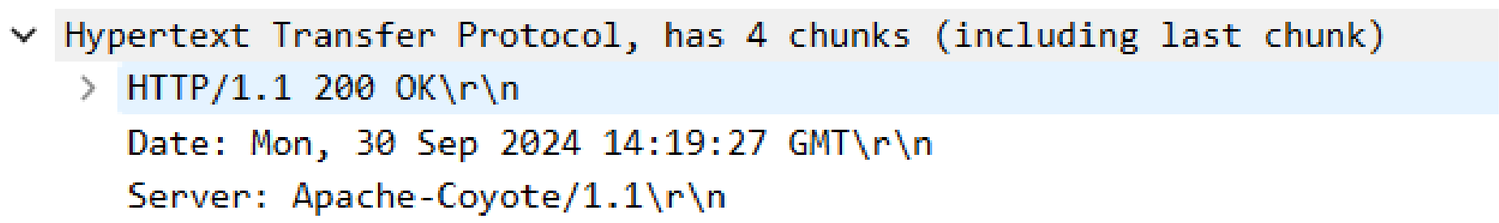
Where, we get:

- Number of segment: 8
- TCP segment data: 7669 Bytes

Part-G

Click on the packet with 200 OK status code, and expand the ***Hypertext Transfer Protocol*** section.

Look for the line that starts with ***HTTP/1.1***, which will be followed by the status code and phrase associated with the response to the HTTP GET request.



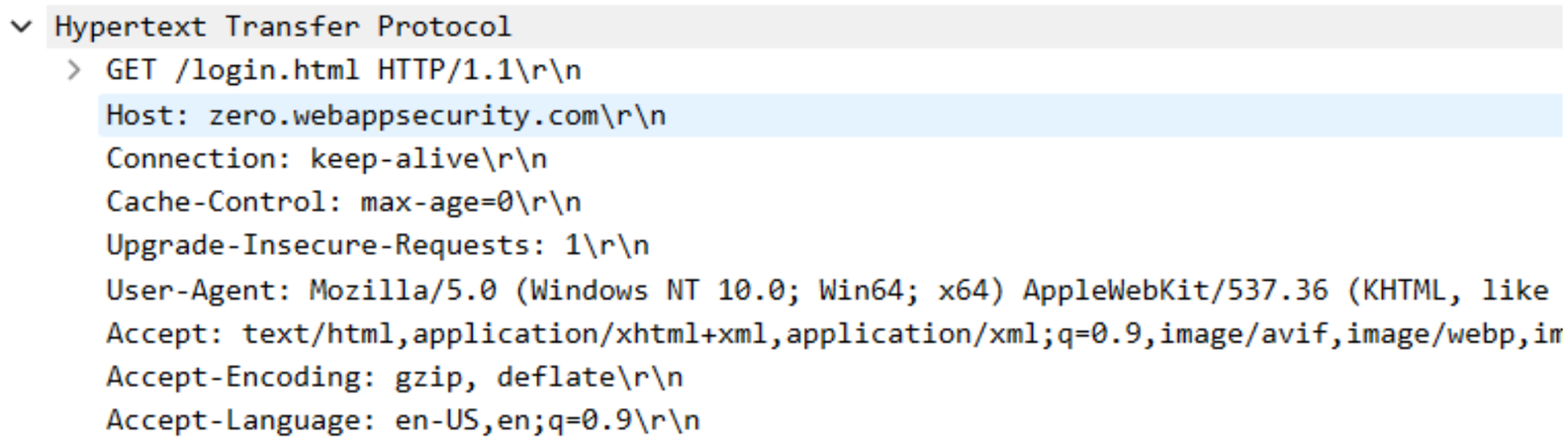
Where, we get:

- Status Code: ***200***
- Status Phrase: ***OK***

Part-H

Find the first HTTP request packet, which has GET method, click on it, then in packet details pane, expand the ***Hypertext Transfer Protocol*** section.

- Look for the Connection header in the HTTP request:
  - If it says ***Connection: keep-alive***, the connection is ***persistent***.
  - If it says ***Connection: close***, the connection is ***non-persistent.***



In my case it says, ***Connection: keep-alive***, so the connection is ***persistent***.