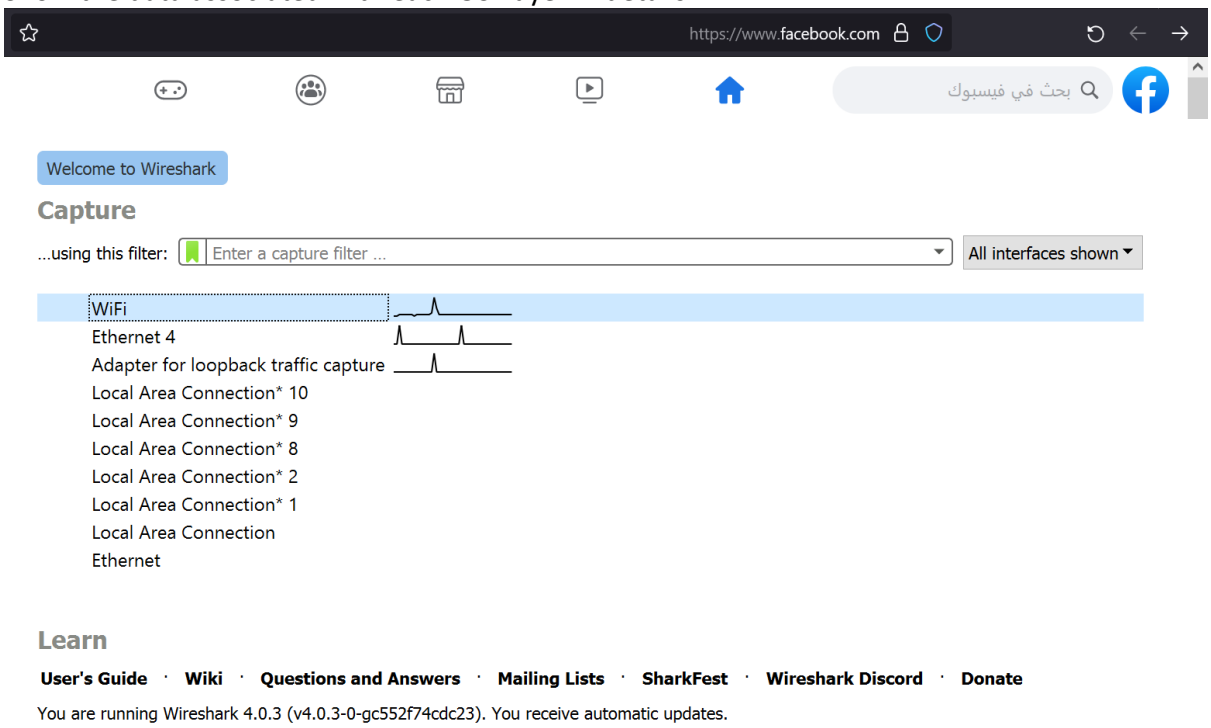
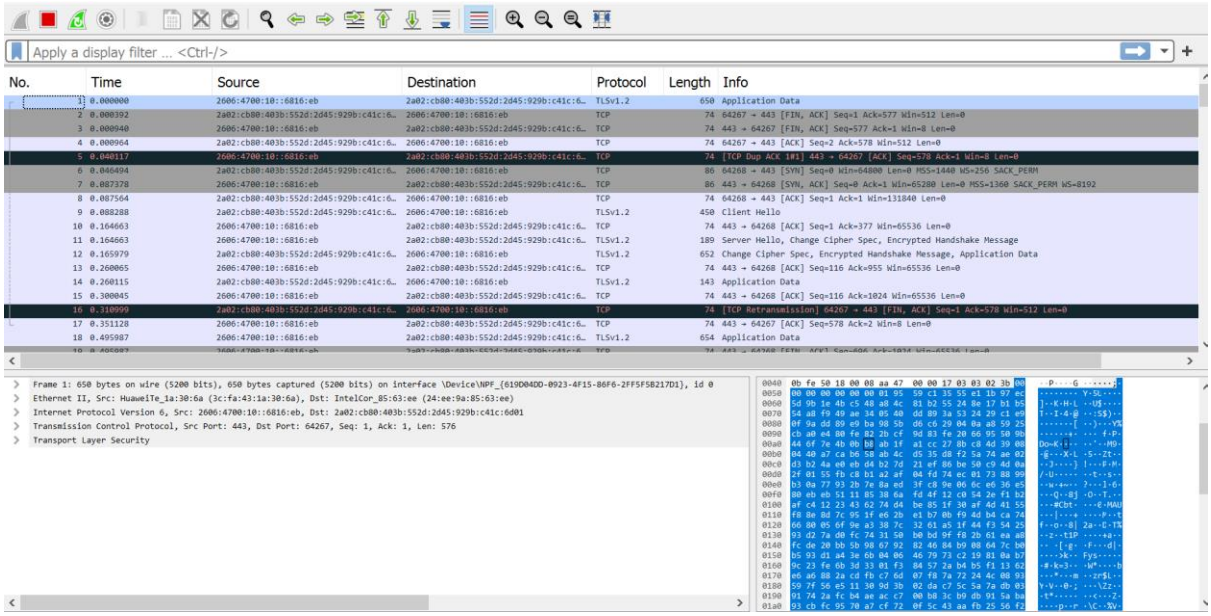
# PROJECT

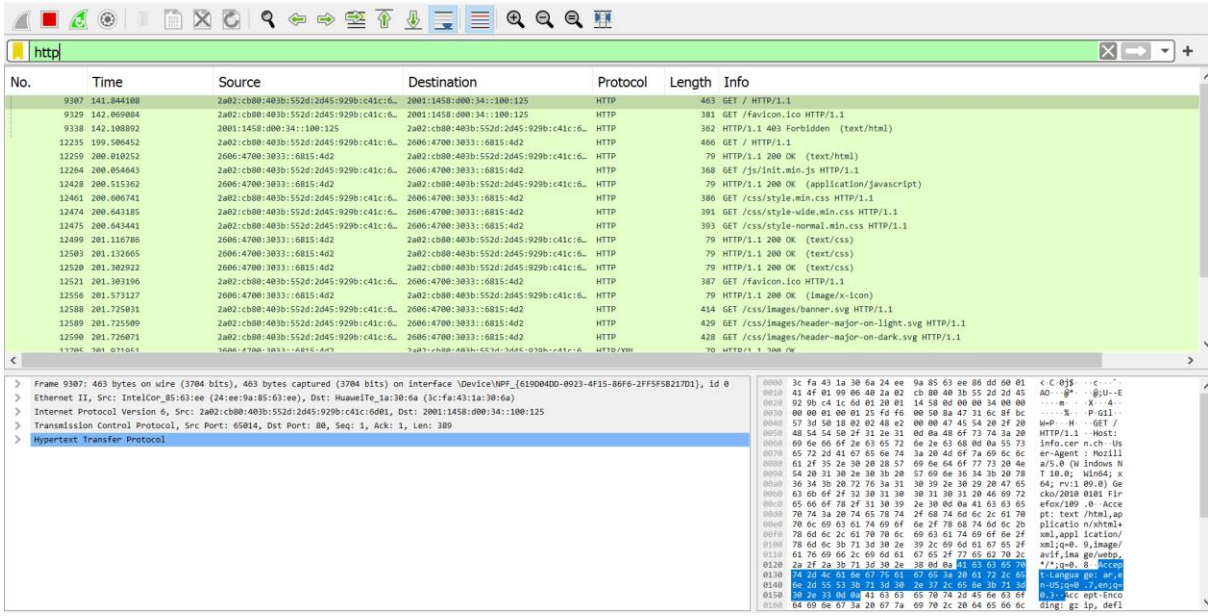
1 . Visit a login page in any trusted website and use Wireshark to : ( 2 Marks)
Show the data associated with each OSI layer in details.



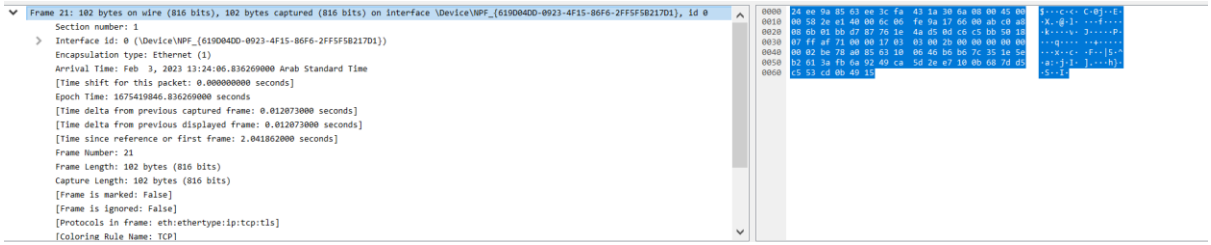

After selecting wifi there are packets captured:
capture the packets:



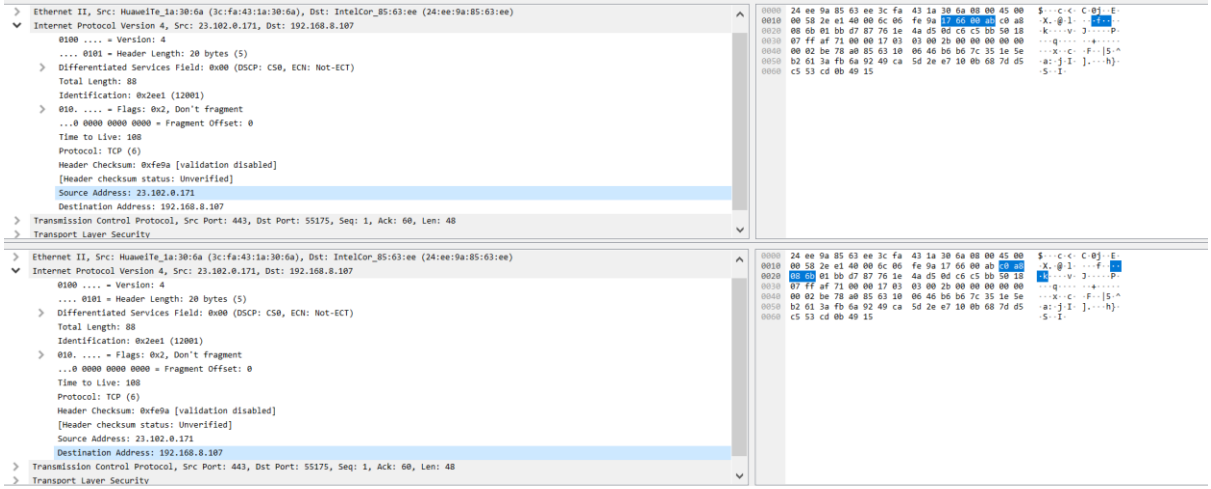
As you can see now with shark:
It start works:




First we have the physical layer ( data transmitted into bits):



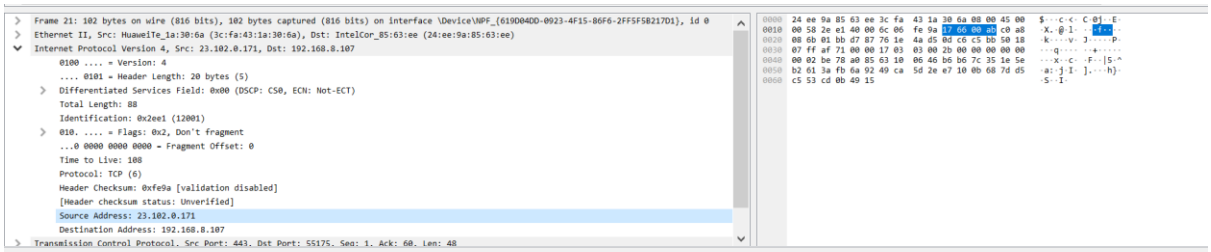

Second we have the Data Link Layer which represent with Ethernet as protocol:
As you can see we have the mac address of source and destination :





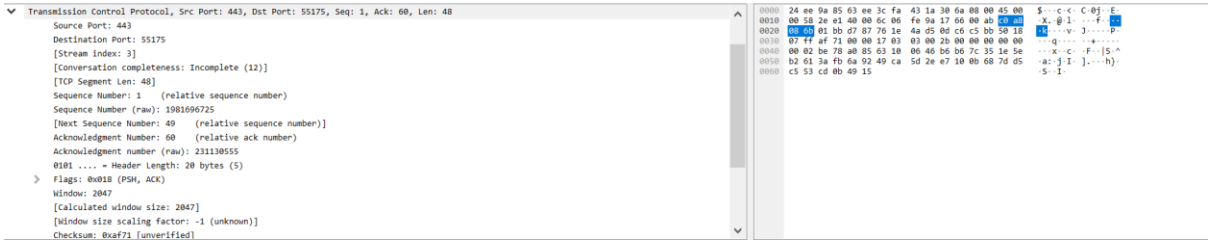


Third we have the Network Layer where we have the IP here we have version 4 also you can see our ip source and destination :

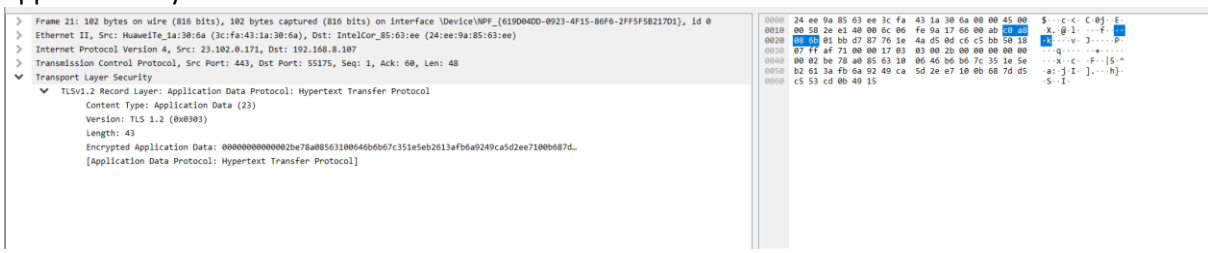




Fourth it's the UDP and TCP layer so it's the Transport layer and here you see the 443 ports and dest 55175 :




And in the last here we can find the session and presentation and application layers so HTTP is the application Layer:



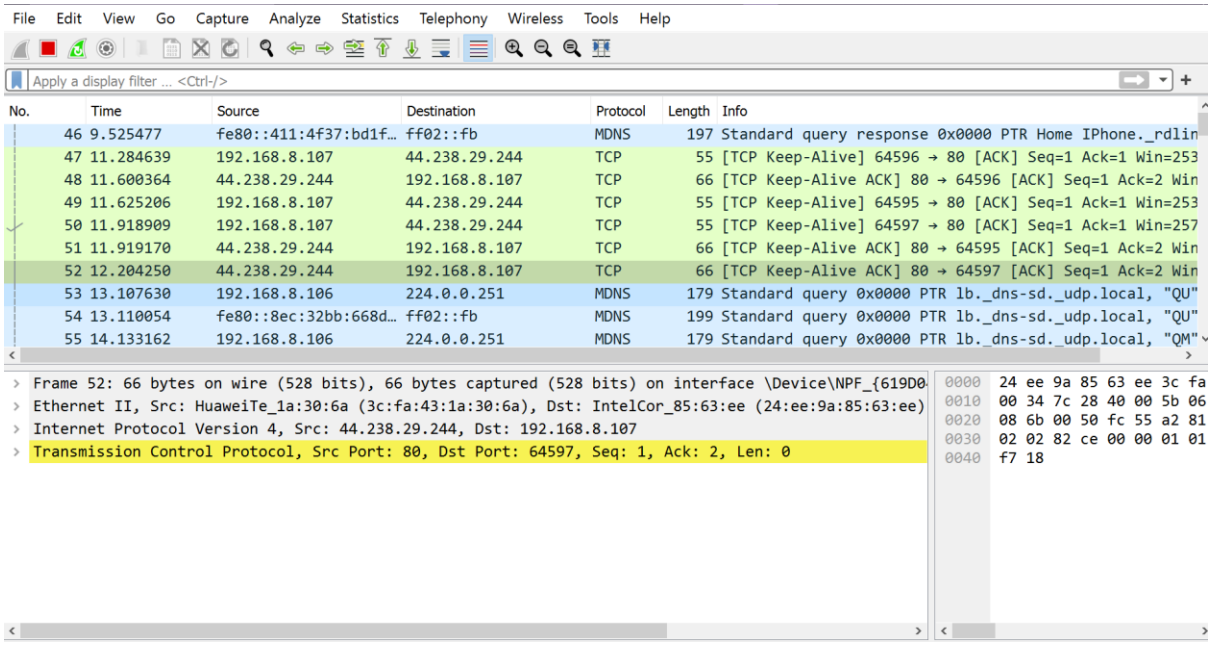

You can see the application is: HTTP and the encrypted for security.

---

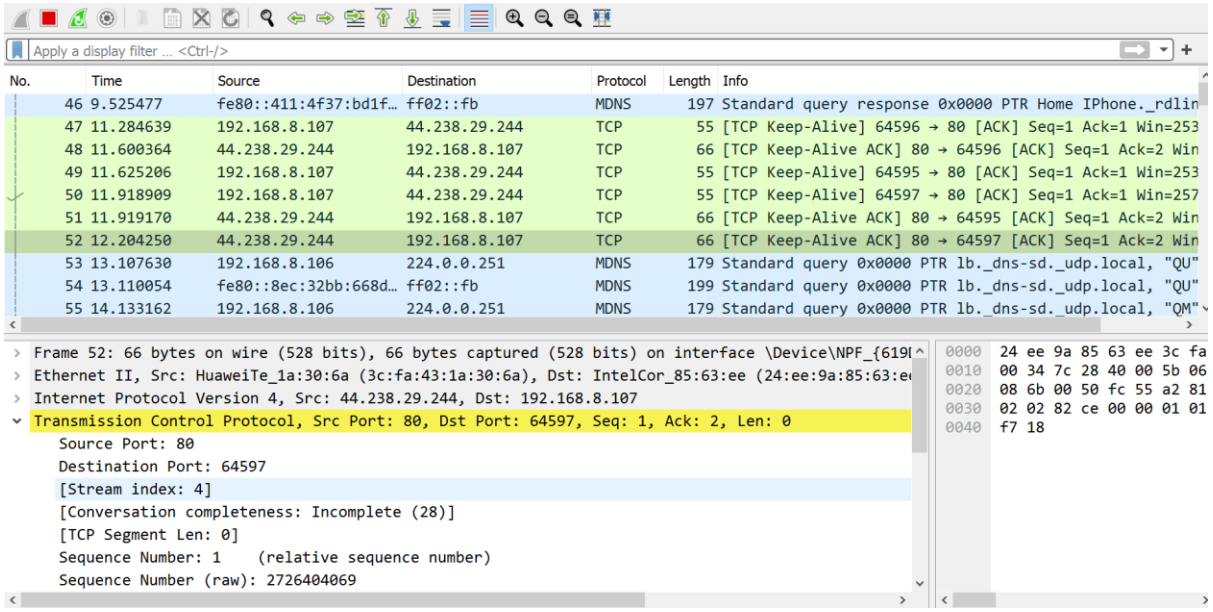
do the same task but with http page (not encrypted) and show the differences between layers.

Here the website for http only

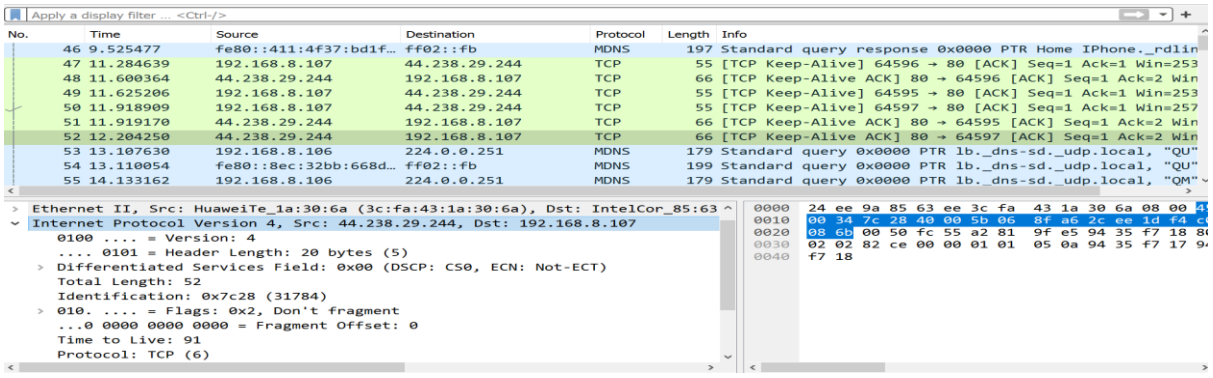
First the https was operates at the Application Layer, whereas HTTPS operates at Transport Layer

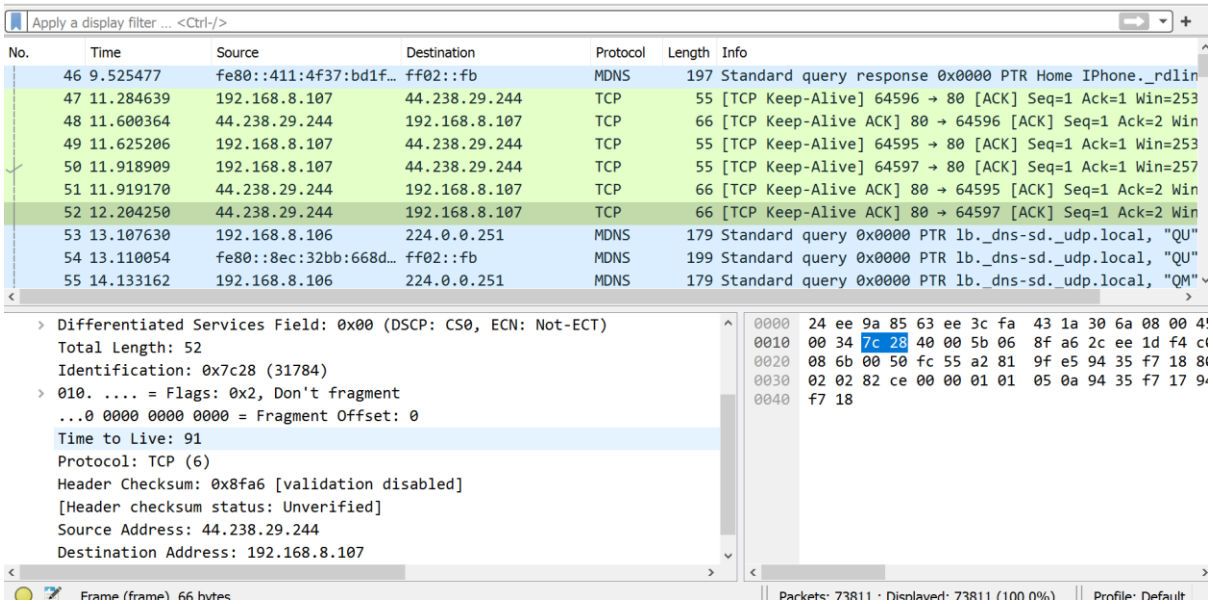



Also we can see here at the transport layer that the port for http(80) in https was(443)

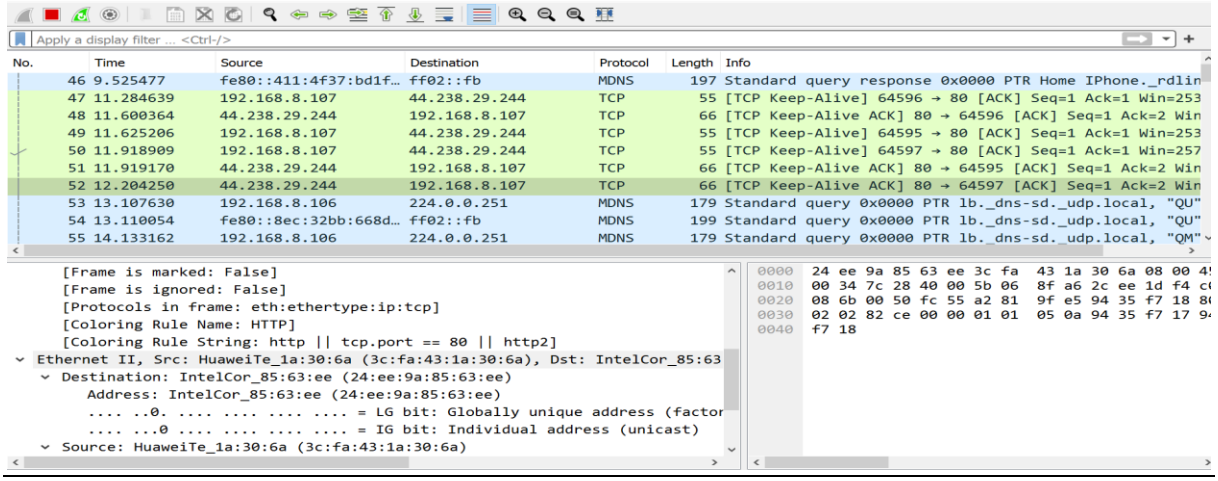



Here we have the network layer for http it operats with TCP/IP not like https operats with TLS






And here we can see the data link layer and the physical layer:

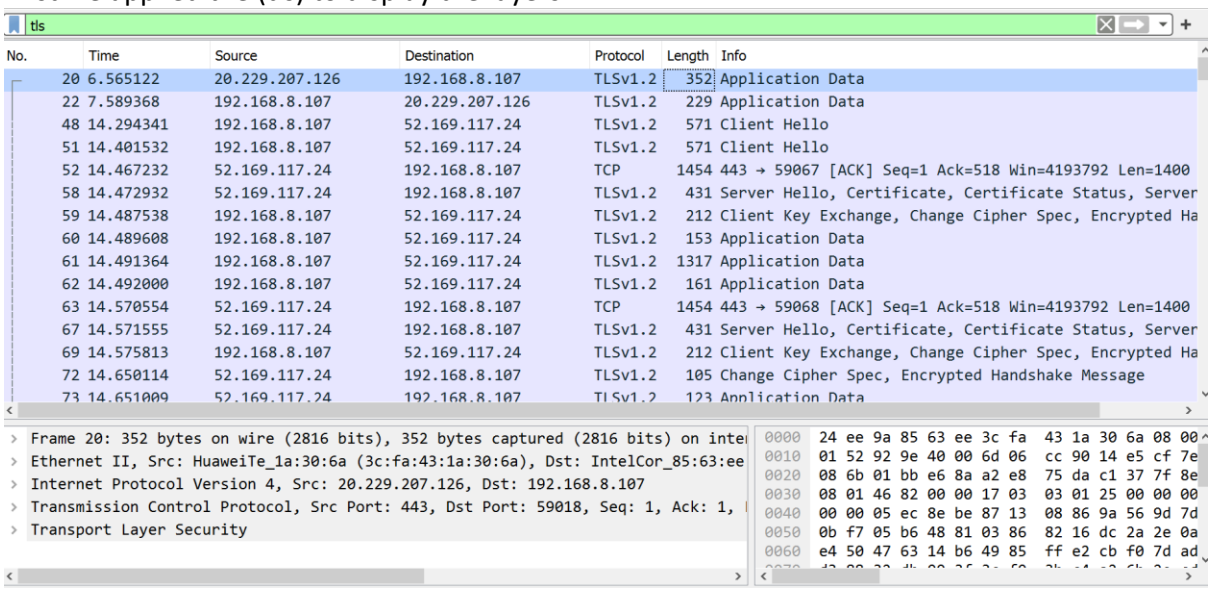


2. Establish a handshake session with any trusted website and show the following :

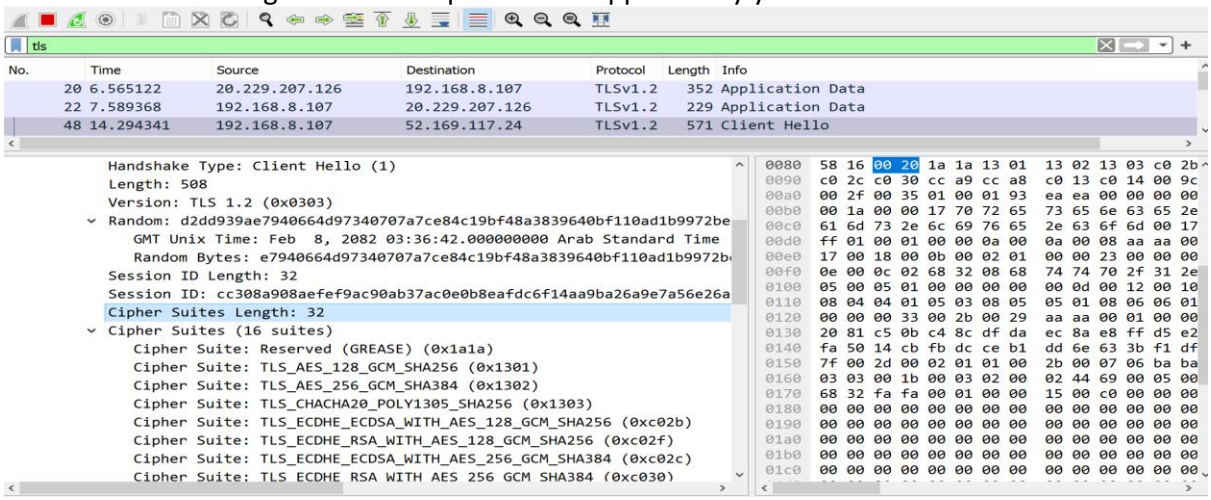
1 . What is the IP address of the website >>use nslookup in command terminal .

```
C:\Users\shama>nslookup www.instagram.com
Server:  UnKnown
Address:  fe80::3efa:43ff:fe1a:306a

Non-authoritative answer:
Name:    z-p42-instagram.c10r.instagram.com
Addresses:  2a03:2880:f242:e0:face:b00c:0:4420
          157.240.195.174
Aliases:  www.instagram.com
          geo-p42.instagram.com


C:\Users\shama>
```

Then from Wireshark, inspect the following:
First we applied the (tls) to display the layers

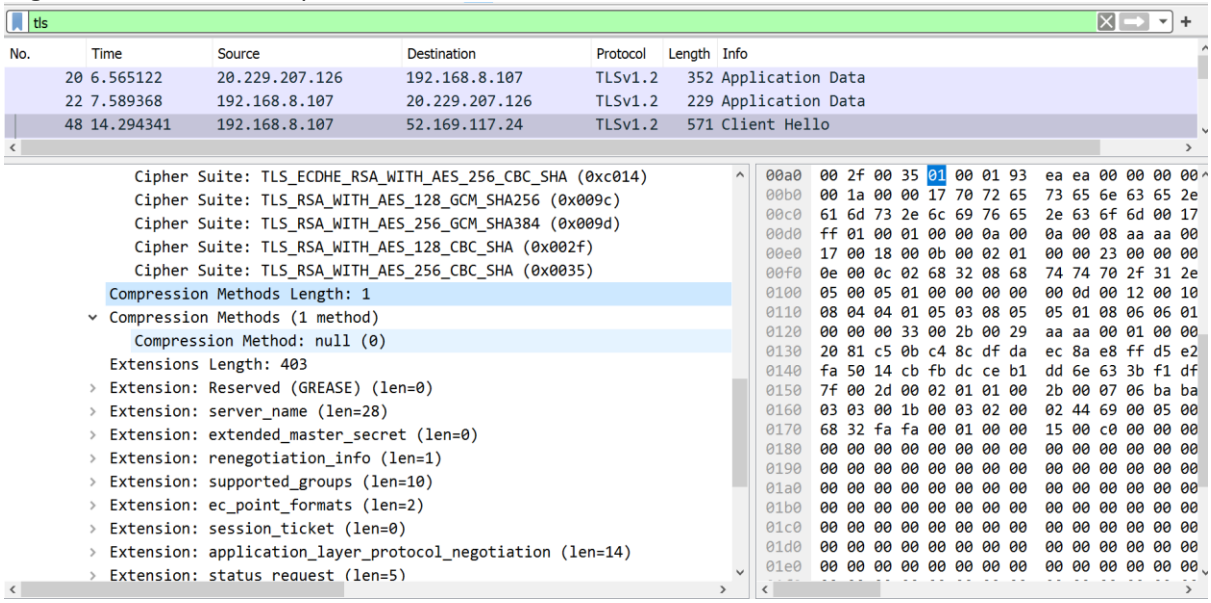



2. Client Hello message and show cipher suite supported by your browser



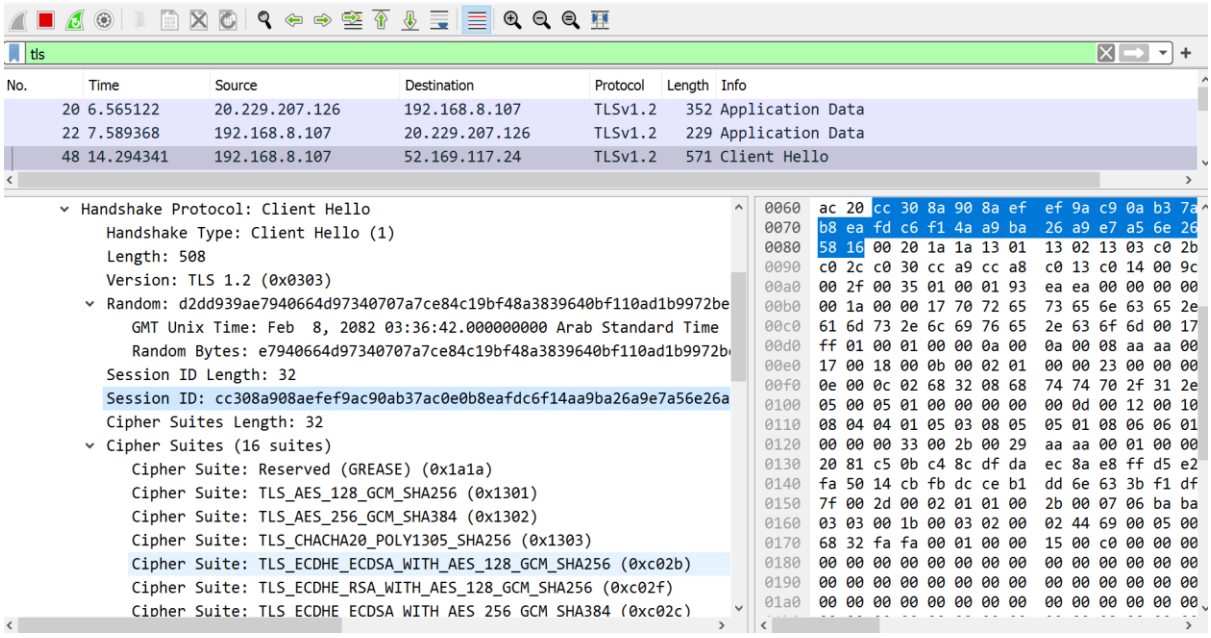
3. What is the compression method?

Algorithms used to compress stored files.



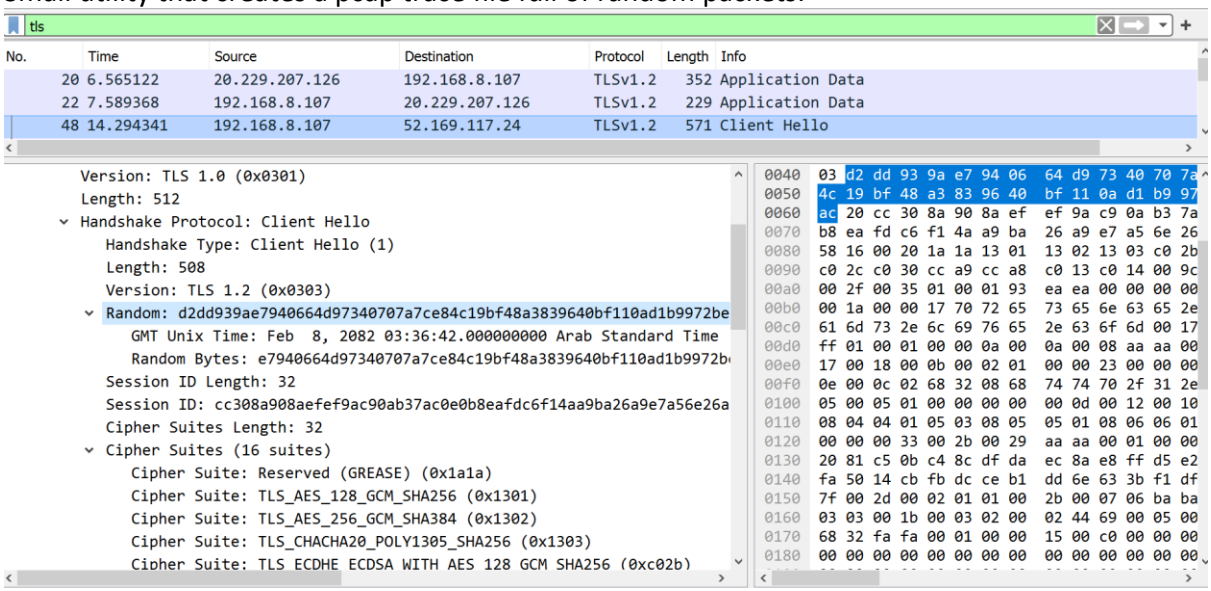

4. What is the session ID?

Unique number from the website to the user in the duration of the visit.

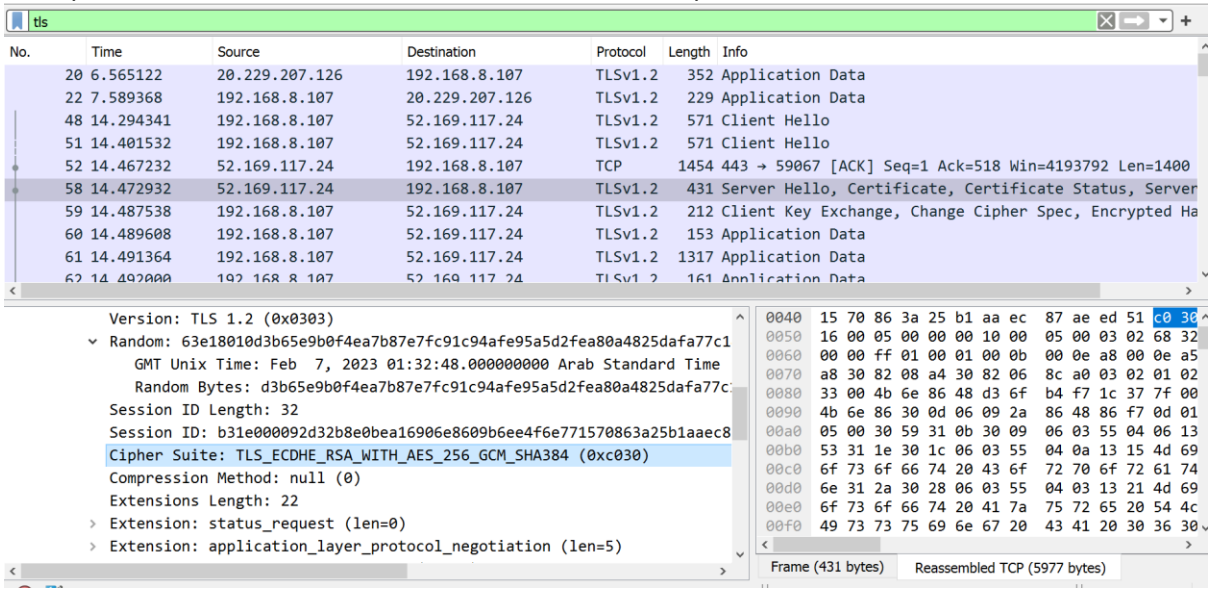


5.What is Random number?

Small utility that creates a pcap trace file full of random packets.




6.Inspect Server Hello, what did the server choose in cipher suite .




Protocol: TLS

Key exchange: ECDHE

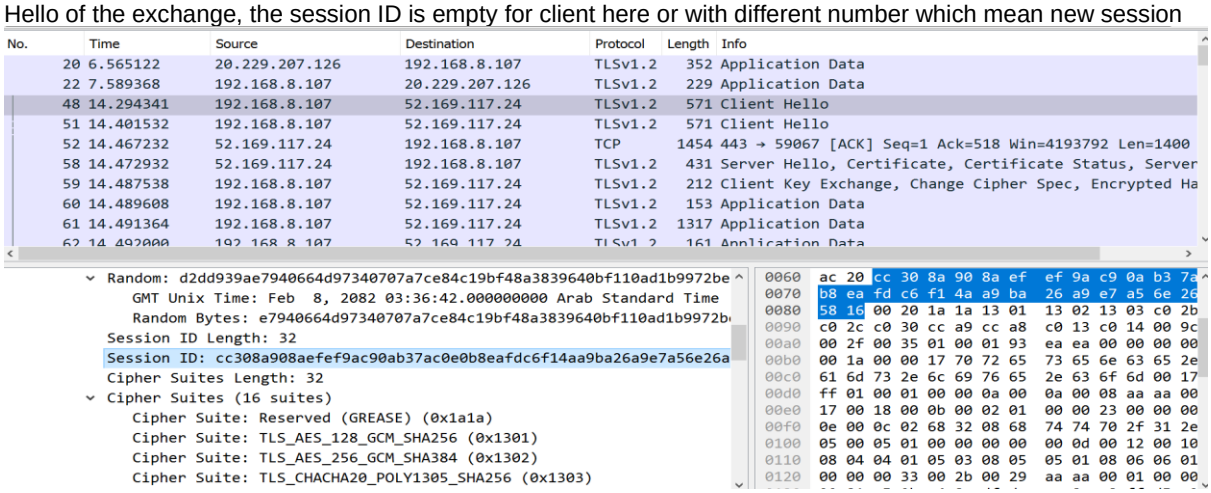
Authentication: RSA

Encrypted: AES 128 GCM

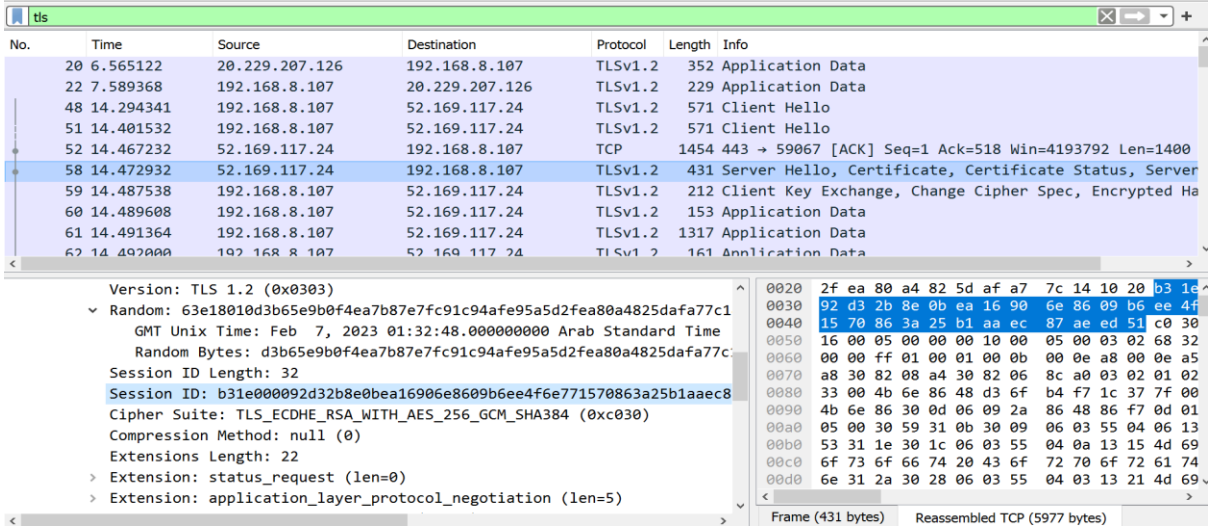
Hash: SHA256

7.Is the session ID the same from Server Hello What does this refer to ?

The ID of a session the client wishes to use for this connection. In the first Client Hello of the exchange, the session ID is empty for client here or with different number which mean new session

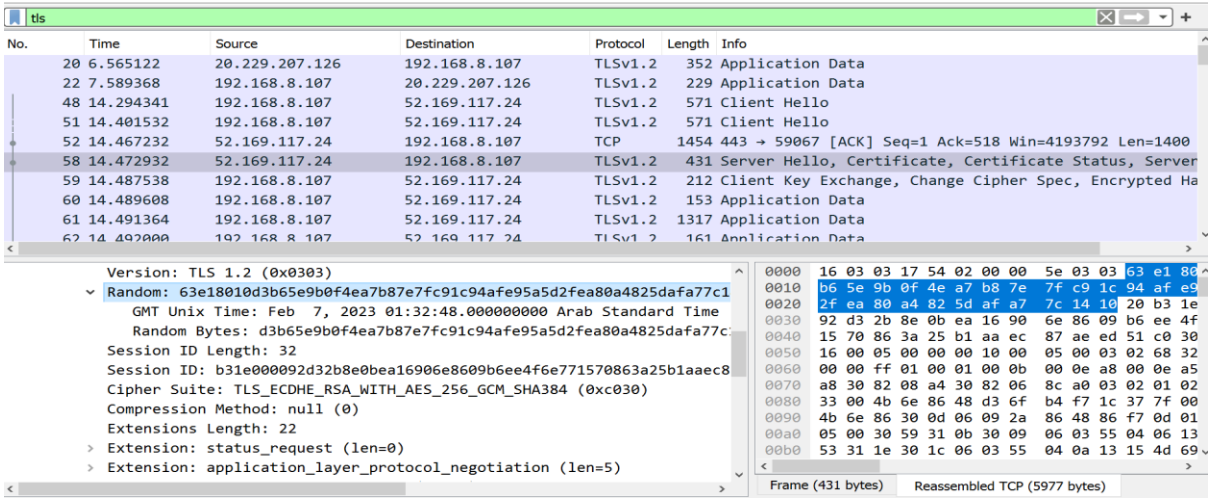



and for server it's: also different because for user it's new session but not recovering a pervious session to be the same number.



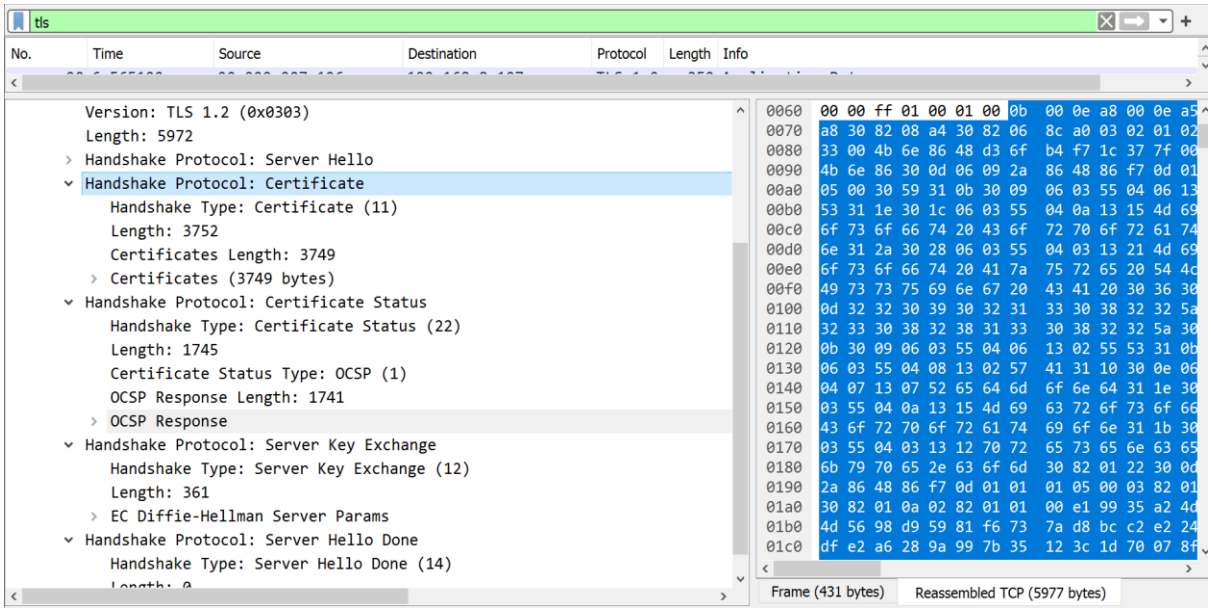

8.Is the Random number the same from Server Hello? Why ?

number is different and why it must be different because later will use this number to generate the key for encryption, by this we approve the message is refresh and the attacker cannot take it and prevent replay attack.

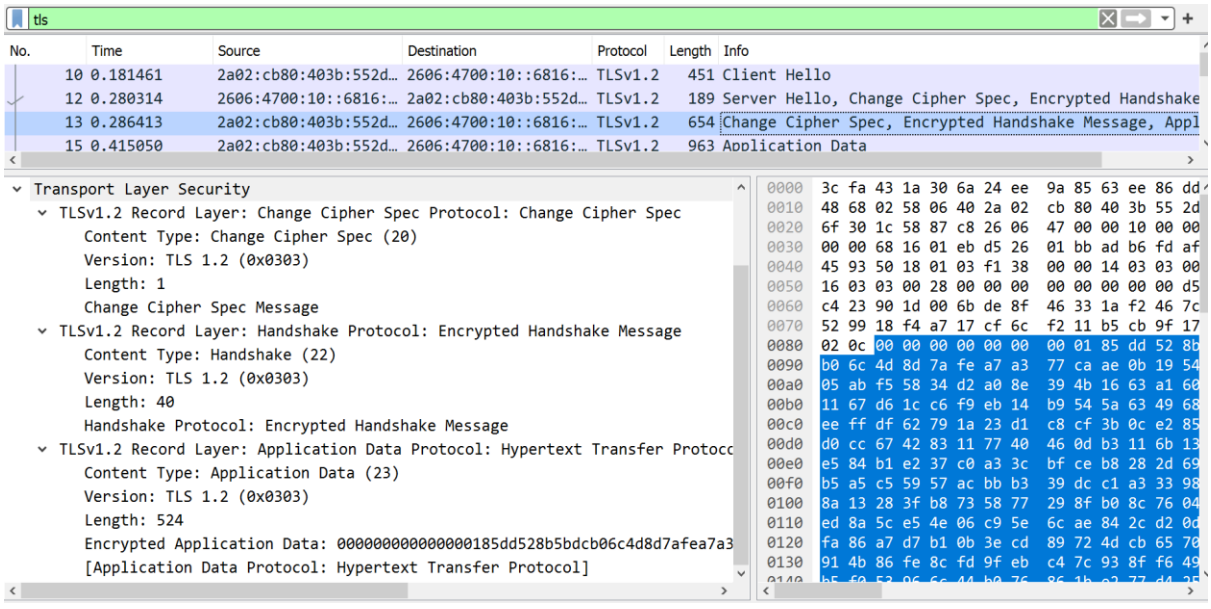


9.Is there a certificate packet between KKU and your browser? Why ?

Yes, to approve the identity of the user that visit the site
Also, to let the user felt safe while visiting the website

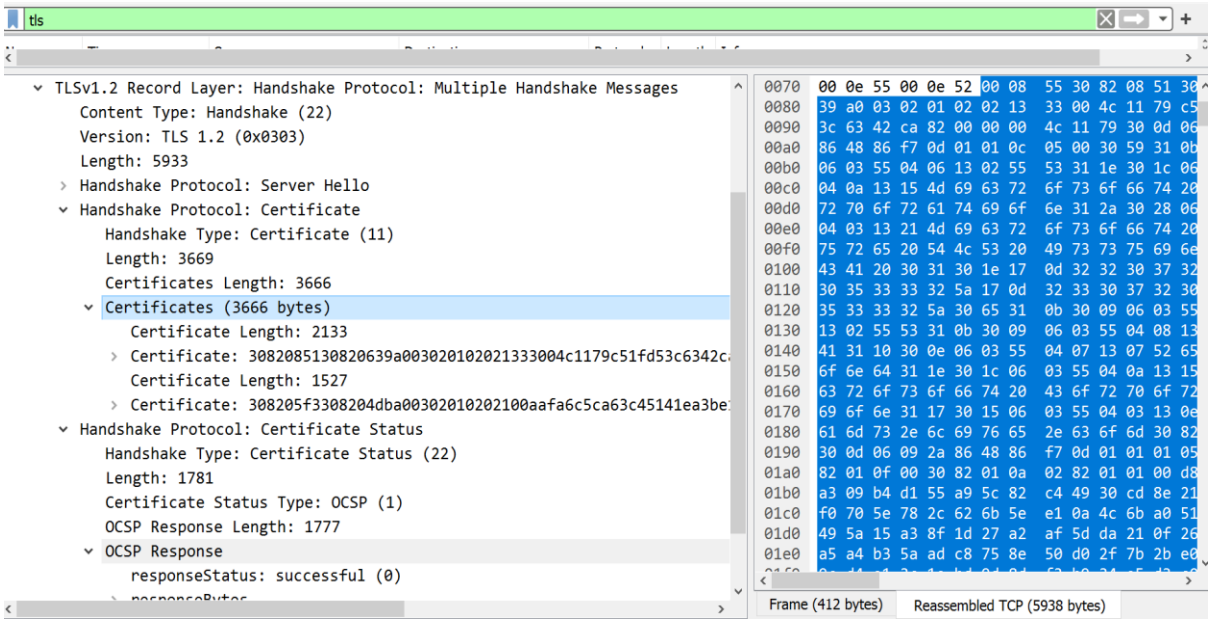



10.Inspect Change cipher spec, do you see the contents of the message? If not, why ?
No , because the message is encrypted



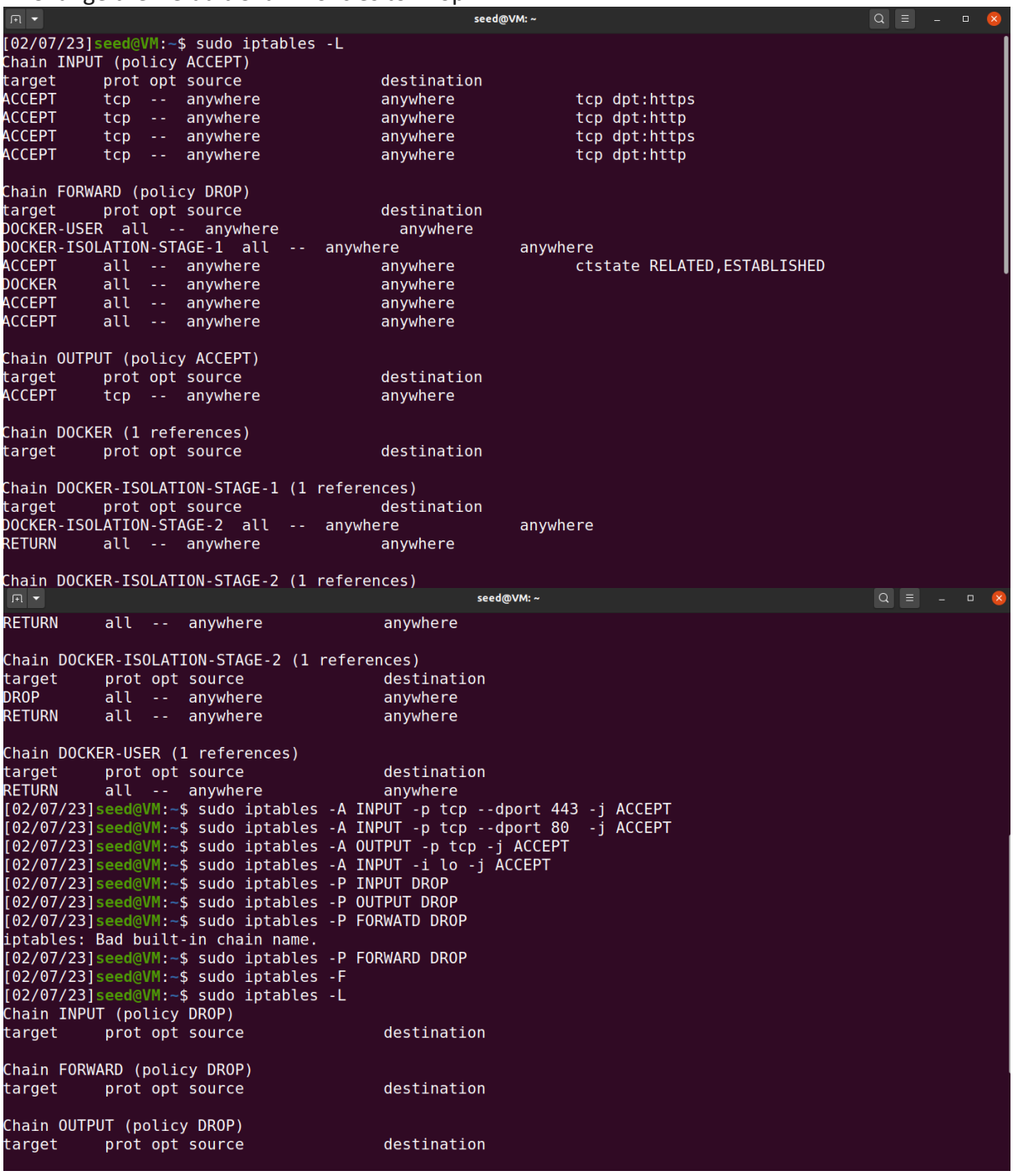
11.Open any other website (https) and show the certificate of the website and explain .its contents captured by Wireshark (BLACKBOARD)


The website is blackboard here I will show the content in the other screenshot:

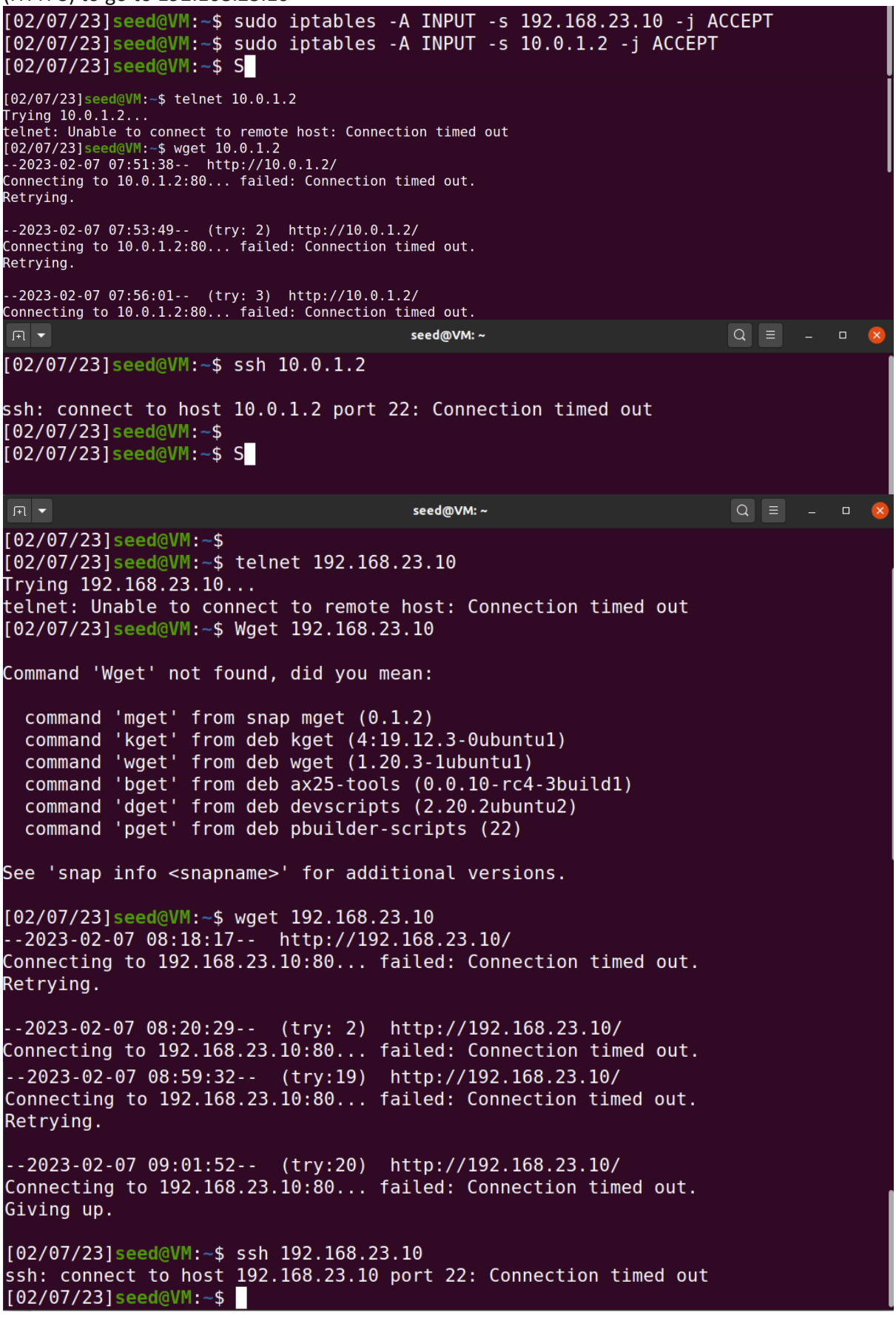


3 . Building a simple Firewall on the client machine :

1-Change the Default Chain Policies to Drop



Allow forwarding of TCP traffic on IP interface 10.0.1.2 (client) port 80 (HTTP) and port 443 (HTTPS) to go to 192.168.23.10



**3-block a specific ip-address from reaching organization.**

```
[02/07/23]seed@VM:~$
[02/07/23]seed@VM:~$
[02/07/23]seed@VM:~$
[02/07/23]seed@VM:~$ sudo iptables -A INPUT -s 10.0.2.15 -j DROP
[02/07/23]seed@VM:~$
```

**4-Allow Incoming HTTP and HTTPS**

To allow HTTP Web traffic:

```
[02/07/23]seed@VM:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
[02/07/23]seed@VM:~$
```

To allow HTTPS Internet traffic:

```
[02/07/23]seed@VM:~$ sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
[02/07/23]seed@VM:~$
[02/07/23]seed@VM:~$
```

**5-**Allow outbound DNS.

```
[02/07/23]seed@VM:~$ sudo iptables -A OUTPUT -p tcp --dport 53 -j ACCEPT
[02/07/23]seed@VM:~$ sudo iptables -A INPUT -p tcp --dport 53 -j ACCEPT
[02/07/23]seed@VM:~$ sudo iptables -A INPUT -p udp --dport 53 -j ACCEPT
[02/07/23]seed@VM:~$ sudo iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
```

DNS connections are made to a remote host on port 53 using the UDP protocol with a fall back to TCP if the answer is too large for a UDP datagram.

**DONE**

---


| NAMES: | ID |
|---|---|
| SHAMAM ALKAFRI | **441813693** |
| RAZAN AHMAD | **441808325** |
| FAYROUZ ASIRI | **439803568** |