

Lab 5 Report

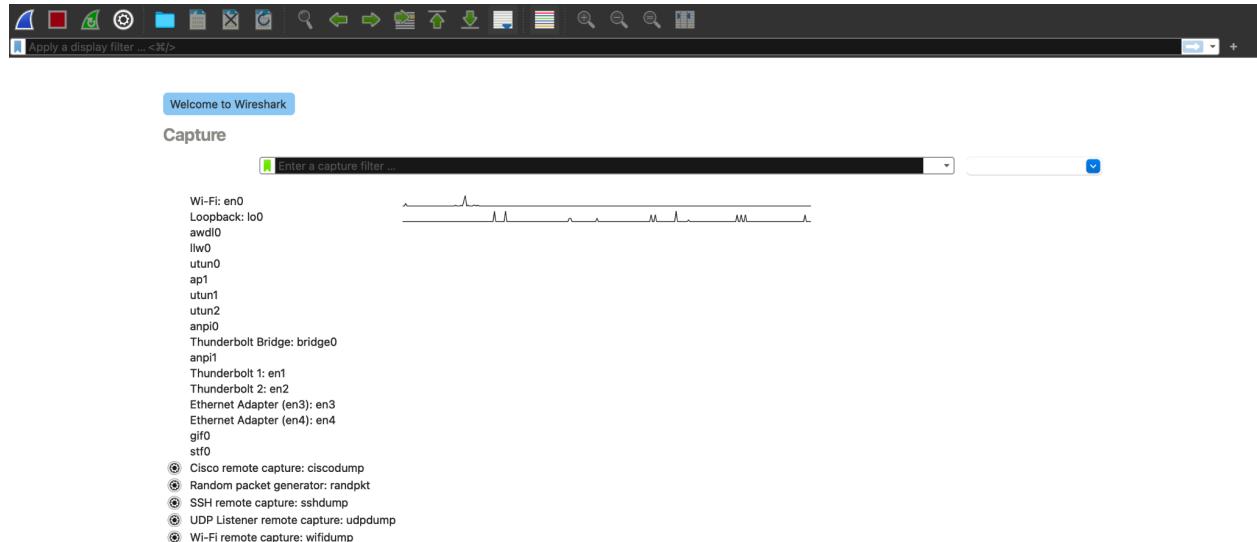
CSL 6010 - Cyber Security

Rahul Barodia

B20CS047

1) Packet Capturing

First, we need to select the wireless interface which we need to observe.



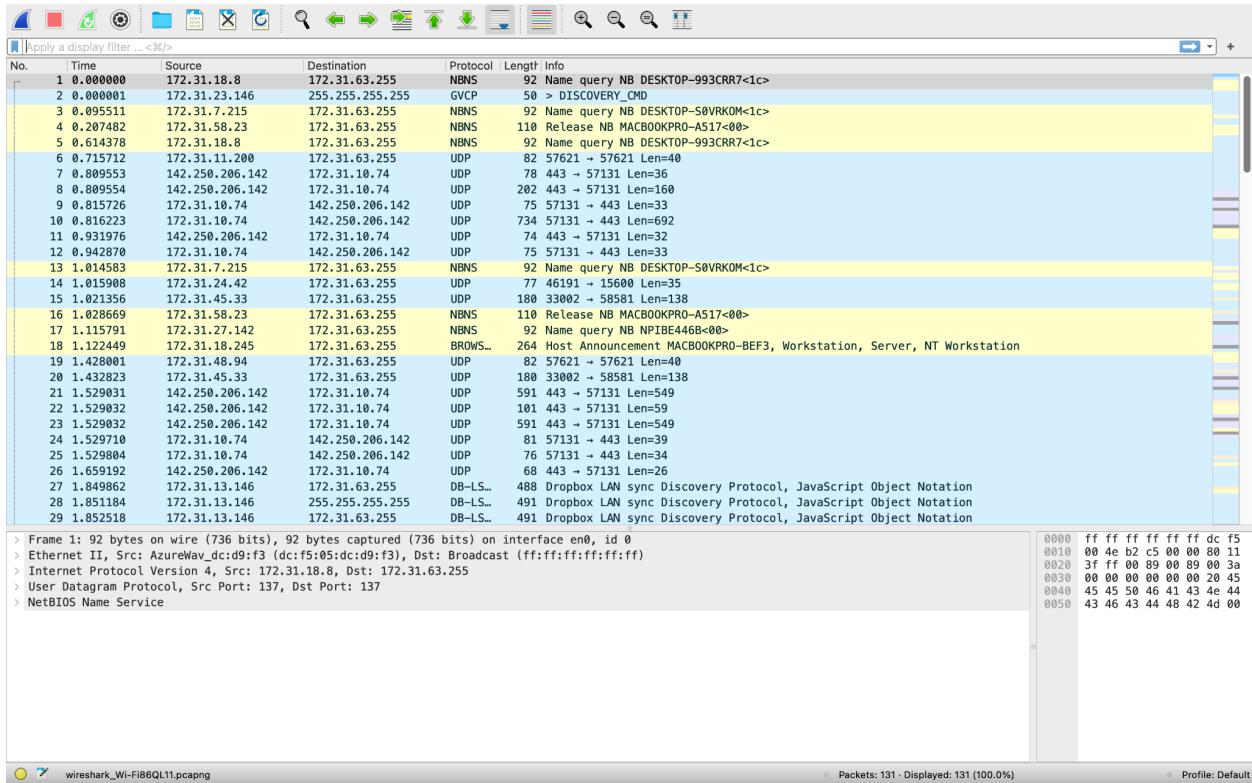
Learn

[User's Guide](#) [Wiki](#) [Questions and Answers](#) [Mailing Lists](#) [SharkFest](#) [Wireshark Discord](#) [Donate](#)



Upon clicking it packet capturing starts and based on our connected network.

I clicked on the Wi-Fi and the following packet were captured:



We can observe time , source and destination , type of protocol and info of each of these packets.

2) Packet capturing upon visiting IITJ website

I started packet capturing and then visited www.iitj.ac.in and then stopped packet capturing.

Upon entering the DNS filter I was able to see the following DNS requests

No.	Time	Source	Destination	Protocol	Length	Info
51	4.246917	172.31.10.74	172.16.100.205	DNS	75	Standard query 0x6129 A www.youtube.com
52	4.247060	172.31.10.74	172.16.100.205	DNS	75	Standard query 0xb6ea HTTPS www.youtube.com
53	4.265392	172.16.100.205	172.31.10.74	DNS	529	Standard query response 0x6129 A www.youtube.com CNAME youtube-ui.l.google.com A 142.250.194.78 A
54	4.265393	172.16.100.205	172.31.10.74	DNS	372	Standard query response 0xb6ea HTTPS www.youtube.com CNAME youtube-ui.l.google.com HTTPS NS ns3.google.com
81	5.627775	172.31.10.74	172.16.100.205	DNS	74	Standard query 0x9287 A www.google.com
82	5.627839	172.31.10.74	172.16.100.205	DNS	74	Standard query 0x77cc HTTPS www.google.com
84	5.693662	172.16.100.205	172.31.10.74	DNS	347	Standard query response 0x77cc HTTPS www.google.com HTTPS NS ns3.google.com NS ns2.google.com NS ns1.google.com
85	5.693663	172.16.100.205	172.31.10.74	DNS	338	Standard query response 0x9287 A www.google.com A 142.250.194.132 NS ns3.google.com NS ns4.google.com
114	6.128301	172.31.10.74	172.16.100.205	DNS	74	Standard query 0xc594 A www.iitj.ac.in
115	6.128530	172.31.10.74	172.16.100.205	DNS	74	Standard query 0x31fe HTTPS www.iitj.ac.in
116	6.154215	172.16.100.205	172.31.10.74	DNS	124	Standard query response 0xc594 A www.iitj.ac.in A 172.16.100.5 NS dns.iitj.ac.in A 172.16.100.205
117	6.154215	172.16.100.205	172.31.10.74	DNS	119	Standard query response 0x31fe HTTPS www.iitj.ac.in SOA dns.iitj.ac.in
125	6.178800	172.31.10.74	172.16.100.205	DNS	76	Standard query 0xfb1 A iitj.ac.in
126	6.178844	172.31.10.74	172.16.100.205	DNS	76	Standard query 0x62dc HTTPS iitj.ac.in
127	6.184575	172.16.100.205	172.31.10.74	DNS	115	Standard query response 0x62dc HTTPS iitj.ac.in SOA dns.iitj.ac.in
128	6.184576	172.16.100.205	172.31.10.74	DNS	120	Standard query response 0xfb1 A iitj.ac.in A 172.16.100.5 NS dns.iitj.ac.in A 172.16.100.205

TCP:

No.	Time	Source	Destination	Protocol	Length	Info
118	6.154451	172.31.10.74	172.16.100.5	TCP	78	59637 - 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSeqr=2023535552 TSeqc=0 SACK_PERM
119	6.168354	172.16.100.5	172.31.10.74	TCP	74	80 - 59637 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM TSeqr=1898733272 TSeqc=2023535552
120	6.168525	172.31.10.74	172.16.100.5	TCP	66	59637 - 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSeqr=2023535566 TSeqc=1898733272
→ 121	6.168590	172.31.10.74	172.16.100.5	HTTP	513	GET / HTTP/1.1
122	6.174003	172.16.100.5	172.31.10.74	TCP	66	80 - 59637 [ACK] Seq=1 Ack=448 Win=30080 Len=0 TSeqr=1898733286 TSeqc=2023535566
→ 123	6.177156	172.16.100.5	172.31.10.74	HTTP	497	HTTP/1.1 302 Found (text/html)
124	6.177194	172.31.10.74	172.16.100.5	TCP	66	59637 - 80 [ACK] Seq=448 Ack=432 Win=131328 Len=0 TSeqr=2023535575 TSeqc=1898733286
8683	11.489286	172.16.100.5	172.31.10.74	TCP	66	80 - 59637 [FIN, ACK] Seq=432 Ack=448 Win=30080 Len=0 TSeqr=1898738294 TSeqc=2023535575
8685	11.489396	172.31.10.74	172.16.100.5	TCP	66	59637 - 80 [ACK] Seq=448 Ack=433 Win=131328 Len=0 TSeqr=2023540887 TSeqc=1898738290
8971	11.640459	172.31.10.74	172.16.100.5	TCP	66	59637 - 80 [FIN, ACK] Seq=448 Ack=433 Win=131328 Len=0 TSeqr=2023541038 TSeqc=1898738290
9888	11.669984	172.16.100.5	172.31.10.74	TCP	66	[TCP Retransmission] 80 - 59637 [FIN, ACK] Seq=432 Ack=448 Win=30080 Len=0 TSeqr=1898738503 TSeqc=2023541038
9895	11.669914	172.31.10.74	172.16.100.5	TCP	66	[TCP Retransmission] 59637 - 80 [FIN, ACK] Seq=448 Ack=433 Win=131328 Len=0 TSeqr=2023541068 TSeqc=1898738503
9389	11.834859	172.16.100.5	172.31.10.74	TCP	66	80 - 59637 [ACK] Seq=433 Ack=449 Win=30080 Len=0 TSeqr=1898738758 TSeqc=2023541038
9454	11.881501	172.16.100.5	172.31.10.74	TCP	66	[TCP Dup ACK 9389#1] 80 - 59637 [ACK] Seq=433 Ack=449 Win=30080 Len=0 TSeqr=1898738788 TSeqc=2023541038
9476	11.881612	172.31.10.74	172.16.100.5	TCP	54	59637 - 80 [RST] Seq=449 Win=0 Len=0

HTTP:

No.	Time	Source	Destination	Protocol	Length	Info
121	6.168590	172.31.10.74	172.16.100.5	HTTP	513	GET / HTTP/1.1
123	6.177156	172.16.100.5	172.31.10.74	HTTP	497	HTTP/1.1 302 Found (text/html)

We can get the IP address of the IITJ server by observing the DNS response packet which contains our IP address

No.	Time	Source	Destination	Protocol	Length	Info
51	4.246917	172.31.10.74	172.16.100.205	DNS	75	Standard query 0x6129 A www.youtube.com
52	4.247060	172.31.10.74	172.16.100.205	DNS	75	Standard query 0xb6ea HTTPS www.youtube.com
53	4.265392	172.16.100.205	172.31.10.74	DNS	529	Standard query response 0x6129 A www.youtube.com CNAME youtube-ui.l.google.com A 142.250.194.78 A
54	4.265393	172.16.100.205	172.31.10.74	DNS	372	Standard query response 0xb6ea HTTPS www.youtube.com CNAME youtube-ui.l.google.com HTTPS NS ns3.google.com
81	5.627775	172.31.10.74	172.16.100.205	DNS	74	Standard query 0x9287 A www.google.com
82	5.627839	172.31.10.74	172.16.100.205	DNS	74	Standard query 0x77cc HTTPS www.google.com
84	5.693662	172.16.100.205	172.31.10.74	DNS	347	Standard query response 0x77cc HTTPS www.google.com HTTPS NS ns3.google.com NS ns2.google.com NS ns1.google.com
85	5.693663	172.16.100.205	172.31.10.74	DNS	338	Standard query response 0x9287 A www.google.com A 142.250.194.132 NS ns3.google.com NS ns4.google.com
114	6.128301	172.31.10.74	172.16.100.205	DNS	74	Standard query 0xc594 A www.iitj.ac.in
115	6.128530	172.31.10.74	172.16.100.205	DNS	74	Standard query 0x31fe HTTPS www.iitj.ac.in
116	6.154215	172.16.100.205	172.31.10.74	DNS	124	Standard query response 0xc594 A www.iitj.ac.in A 172.16.100.5 NS dns.iitj.ac.in A 172.16.100.205
117	6.154215	172.16.100.205	172.31.10.74	DNS	119	Standard query response 0x31fe HTTPS www.iitj.ac.in SOA dns.iitj.ac.in
125	6.178800	172.31.10.74	172.16.100.205	DNS	76	Standard query 0xfb1 A iitj.ac.in
126	6.178844	172.31.10.74	172.16.100.205	DNS	76	Standard query 0x62dc HTTPS iitj.ac.in
127	6.184575	172.16.100.205	172.31.10.74	DNS	115	Standard query response 0x62dc HTTPS iitj.ac.in SOA dns.iitj.ac.in
128	6.184576	172.16.100.205	172.31.10.74	DNS	120	Standard query response 0xfb1 A iitj.ac.in A 172.16.100.5 NS dns.iitj.ac.in A 172.16.100.205

From No. 116 we can see that IP address of IITJ server is 172.16.100.5

Yes, I can see different HTTP requests/response

No.	Time	Source	Destination	Protocol	Length	Info
121	6.168590	172.31.10.74	172.16.100.5	HTTP	513	GET / HTTP/1.1
123	6.177156	172.16.100.5	172.31.10.74	HTTP	497	HTTP/1.1 302 Found (text/html)

Upon browsing on google , I learned that HTTP GET request is being sent to the server if the user scrolls the homepage of the website. I scrolled the IITJ website so I got HTTP GET request followed by HTTP response containing HTML code for the homepage.

3) Packet highlighted in Black Colour

Packet highlighted in Black colour signifies that the packet contains some error.

4) Filters in Wireshark

Filters are used so that the users can see particular network traffic.

I can see the following filters in Wireshark

Filter Name	Filter Expression
Ethernet address 00:00:5e:00:53:00	ether host 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	ether proto 0x0806
No Broadcast and no Multicast	not broadcast and not multicast
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	host 192.0.2.1
IPv6 only	ip6
IPv6 address 2001:db8::1	host 2001:db8::1
TCP only	tcp
UDP only	udp
Non-DNS	not port 53
TCP or UDP port 80 (HTTP)	port 80
No ARP and no DNS	not arp and port not 53
Non-HTTP and non-SMTP to/from www.wireshark.org	not port 80 and not port 25 and host ...

We can also add Filters by specifying the filter name and the filter expression

The 5 filters that I explored are:

1) http filter

No.	Time	Source	Destination	Protocol	Length	Info
159	5.872999	172.31.10.74	172.16.100.5	HTTP	513	GET / HTTP/1.1
161	5.876530	172.16.100.5	172.31.10.74	HTTP	497	HTTP/1.1 302 Found (text/html)

2) tcp filter

No.	Time	Source	Destination	Protocol	Length	Info
38	3.061045	172.31.10.74	13.224.22.91	TCP	78	61932 -> 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=4075953765 TSecr=0 SACK_PERM
40	3.088936	172.31.10.74	13.224.22.91	TCP	66	61932 -> 443 [ACK] Seq=1 Ack=1 Win=131328 Len=0 TStamp=4075953785 TSecr=2315330459
41	3.081042	172.31.10.74	13.224.22.91	TLSv1_	583	Client Hello
44	3.102325	172.31.10.74	13.224.22.91	TCP	66	61932 -> 443 [ACK] Seq=518 Ack=235 Win=131136 Len=0 TStamp=4075953806 TSecr=2315330480
45	3.103448	172.31.10.74	13.224.22.91	TLSv1_	130	Change Cipher Spec, Application Data
46	3.103530	172.31.10.74	13.224.22.91	TLSv1_	164	Application Data
47	3.103593	172.31.10.74	13.224.22.91	TLSv1_	384	Application Data
52	3.123501	172.31.10.74	13.224.22.91	TCP	66	61932 -> 443 [ACK] Seq=998 Ack=820 Win=130496 Len=0 TStamp=4075953827 TSecr=2315330503
53	3.123819	172.31.10.74	13.224.22.91	TLSv1_	97	Application Data
54	3.123851	172.31.10.74	13.224.22.91	TLSv1_	101	Application Data
156	5.868453	172.31.10.74	172.16.100.5	TCP	78	61934 -> 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TStamp=4017337752 TSecr=0 SACK_PERM
158	5.872916	172.31.10.74	172.16.100.5	TCP	66	61934 -> 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TStamp=4017337758 TSecr=2013683027
159	5.872999	172.31.10.74	172.16.100.5	HTTP	513	GET / HTTP/1.1
162	5.876613	172.31.10.74	172.16.100.5	TCP	66	61934 -> 80 [ACK] Seq=448 Ack=432 Win=131328 Len=0 TStamp=4017337762 TSecr=2013683032

3) udp filter

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.31.10.74	142.250.206.142	UDP	76	55865 -> 443 Len=34
2	0.032596	142.250.206.142	172.31.10.74	UDP	68	443 -> 55865 Len=26
3	1.183426	142.250.194.138	172.31.10.74	UDP	1288	443 -> 64376 Len=1246
4	1.183427	142.250.194.138	172.31.10.74	UDP	1292	443 -> 64376 Len=1250
5	1.183428	142.250.194.138	172.31.10.74	UDP	140	443 -> 64376 Len=98
6	1.199141	172.31.10.74	142.250.194.138	UDP	77	64376 -> 443 Len=35
7	1.390369	172.31.10.74	142.250.194.138	UDP	75	64376 -> 443 Len=33
8	1.418760	142.250.194.138	172.31.10.74	UDP	68	443 -> 64376 Len=26
9	1.625537	172.31.10.74	142.250.194.138	UDP	75	64376 -> 443 Len=33
10	1.648719	142.250.194.138	172.31.10.74	UDP	68	443 -> 64376 Len=26
11	1.856012	172.31.10.74	142.250.194.138	UDP	75	64376 -> 443 Len=33
12	1.879623	142.250.194.138	172.31.10.74	UDP	68	443 -> 64376 Len=26
13	2.089352	172.31.10.74	142.250.194.138	UDP	75	64376 -> 443 Len=33
14	2.113144	142.250.194.138	172.31.10.74	UDP	68	443 -> 64376 Len=26
16	2.320406	172.31.10.74	142.250.194.138	UDP	75	64376 -> 443 Len=33
17	2.344688	142.250.194.138	172.31.10.74	UDP	68	443 -> 64376 Len=26
18	2.548838	172.31.10.74	142.250.194.138	UDP	75	64376 -> 443 Len=33
19	2.572754	142.250.194.138	172.31.10.74	UDP	68	443 -> 64376 Len=26
22	2.985379	172.31.10.74	142.250.194.138	UDP	75	64376 -> 443 Len=33
23	3.028178	142.250.194.138	172.31.10.74	UDP	68	443 -> 64376 Len=26
24	3.837194	172.31.10.74	142.250.194.138	UDP	75	64376 -> 443 Len=33
25	3.868892	142.250.194.138	172.31.10.74	UDP	68	443 -> 64376 Len=26
27	4.931062	172.31.10.74	172.16.100.285	DNS	74	Standard query 0xffffd A www.google.com
28	4.931240	172.31.10.74	172.16.100.205	DNS	74	Standard query 0xf9ac HTTPS www.google.com
29	4.948425	172.16.100.205	172.31.10.74	DNS	338	Standard query response 0xffffd A www.google.com A 142.250.194.132 NS ns1.google.com NS ns2.google.com
30	4.945752	172.16.100.205	172.31.10.74	DNS	347	Standard query response 0xf9ac HTTPS www.google.com HTTPS NS ns4.google.com NS ns3.google.com NS ns5.google.com PDKID=94358cc1d7e6999e, PKN: 1, CRYPTO, PADDING, CRYPTO, CRYPTO, CRYPTO, PING, CRYPTO, PI
31	4.946276	172.31.10.74	142.250.194.132	QUIC	1292	Initial, DCID=94358cc1d7e6999e

4) ICMP filter

No.	Time	Source	Destination	Protocol	Length	Info
1051	9.101451	172.31.10.74	172.16.100.205	ICMP	70	Destination unreachable (Port unreachable)

5) DNS Filter

No.	Time	Source	Destination	Protocol	Length	Info
27	4.931062	172.31.10.74	172.16.100.205	DNS	74	Standard query 0xfdff A www.google.com
28	4.931240	172.31.10.74	172.16.100.205	DNS	74	Standard query 0xf9ac HTTPS www.google.com
29	4.940425	172.16.100.205	172.31.10.74	DNS	338	Standard query response 0xfdff A www.google.com A 142.250.194.132 NS ns1.google.com NS ns2.google.com NS ns3.google.com NS ns4.google.com NS ns5.google.com NS ns6.google.com
30	4.945752	172.16.100.205	172.31.10.74	DNS	347	Standard query response 0xf9ac HTTPS www.google.com HTTPS NS ns4.google.com NS ns3.google.com NS ns2.google.com
58	7.154583	172.31.10.74	172.16.100.205	DNS	88	Standard query 0x2c46 A www.espnrcricinfo.com
59	7.154656	172.31.10.74	172.16.100.205	DNS	88	Standard query 0x7f91 HTTPS www.espnrcricinfo.com
68	7.216737	172.16.100.205	172.31.10.74	DNS	528	Standard query response 0x2c46 A www.espnrcricinfo.com CNAME wildcard-espnrcricinfo.edgekey.net CNAME e333
61	7.219893	172.16.100.205	172.31.10.74	DNS	225	Standard query response 0x7f91 HTTPS www.espnrcricinfo.com CNAME wildcard-espnrcricinfo.edgekey.net CNAME
145	7.757892	172.31.10.74	172.16.100.205	DNS	72	Standard query 0xbdc1 A dcf.espn.com
146	7.757924	172.31.10.74	172.16.100.205	DNS	72	Standard query 0xc300 HTTPS dcf.espn.com
151	7.765523	172.31.10.74	172.16.100.205	DNS	79	Standard query 0x7cd5 A wasssets.hscicdn.com
152	7.765649	172.31.10.74	172.16.100.205	DNS	79	Standard query 0xcb84 HTTPS wasssets.hscicdn.com
153	7.769473	172.31.10.74	172.16.100.205	DNS	76	Standard query 0xd618 A img1.hscicdn.com
154	7.769539	172.31.10.74	172.16.100.205	DNS	76	Standard query 0x2c0a HTTPS img1.hscicdn.com
179	7.842189	172.16.100.205	172.31.10.74	DNS	519	Standard query response 0x7cd5 A wasssets.hscicdn.com CNAME wildcard-hscicdn-ion.edgekey.net CNAME e13225
188	7.846134	172.16.100.205	172.31.10.74	DNS	217	Standard query response 0x2c0a HTTPS img1.hscicdn.com CNAME wildcard-hscicdn.edgekey.net CNAME e132258
181	7.848316	172.16.100.205	172.31.10.74	DNS	512	Standard query response 0xd618 A img1.hscicdn.com CNAME wildcard-hscicdn.edgekey.net CNAME e132258.dsrb
182	7.848317	172.16.100.205	172.31.10.74	DNS	224	Standard query response 0xcb84 HTTPS wasssets.hscicdn.com CNAME wildcard-hscicdn-ion.edgekey.net CNAME e132258
185	7.890231	172.16.100.205	172.31.10.74	DNS	437	Standard query response 0xbdc1 A dcf.espn.com CNAME twdc-dtci.edge.nc0.co CNAME edge-geo.nc0.co A 3.106
186	7.890233	172.16.100.205	172.31.10.74	DNS	186	Standard query response 0xc300 HTTPS dcf.espn.com CNAME twdc-dtci.edge.nc0.co CNAME edge-geo.nc0.co SOA
1018	9.020528	172.31.10.74	172.16.100.205	DNS	87	Standard query 0xdd0e A safebrowsing.googleapis.com
1019	9.020602	172.31.10.74	172.16.100.205	DNS	87	Standard query 0xab36 HTTPS safebrowsing.googleapis.com
1021	9.023349	172.16.100.205	172.31.10.74	DNS	358	Standard query response 0xdd0e A safebrowsing.googleapis.com A 172.217.166.234 NS ns2.google.com NS ns1.google.com
1047	9.088076	172.31.10.74	172.16.100.205	DNS	86	Standard query 0xe098 A s3-eu-west-1.amazonaws.com
1048	9.088110	172.31.10.74	172.16.100.205	DNS	86	Standard query 0xc4fa HTTPS s3-eu-west-1.amazonaws.com
1049	9.091836	172.16.100.205	172.31.10.74	DNS	178	Standard query response 0xc4fa HTTPS s3-eu-west-1.amazonaws.com SOA ns-1546.awsdns-01.co.uk
1050	9.101341	172.16.100.205	172.31.10.74	DNS	144	Standard query response 0x4a36 HTTPS safebrowsing.googleapis.com SOA ns1.google.com
1052	9.118569	172.16.100.205	172.31.10.74	DNS	415	Standard query response 0xe098 A s3-eu-west-1.amazonaws.com A 52.218.109.39 A 52.218.122.32 A 52.218.25
1088	9.509703	172.31.10.74	172.16.100.205	DNS	90	Standard query 0xa0f4 A ver nemqingress.hs-cricinfo.com
1089	9.509771	172.31.10.74	172.16.100.205	DNS	98	Standard query 0xd99f HTTPS ver nemqingress.hs-cricinfo.com
1090	9.513388	172.16.100.205	172.31.10.74	DNS	171	Standard query response 0xd99f HTTPS ver nemqingress.hs-cricinfo.com SOA ns-880.awsdns-46.net
1091	9.559586	172.16.100.205	172.31.10.74	DNS	339	Standard query response 0xa0f4 A ver nemqingress.hs-cricinfo.com A 54.179.167.169 A 52.76.223.9 A 54.251

5) Filter command for all outgoing traffic

We can use the following command to list all the outgoing traffic :

`ip.src == IP address`

In my case, the command is , `ip.src == 172.31.10.74`

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.31.10.74	142.250.206.142	UDP	76	55865 → 443 Len=34
6	1.199141	172.31.10.74	142.250.194.138	UDP	77	64376 → 443 Len=35
7	1.398369	172.31.10.74	142.250.194.138	UDP	75	64376 → 443 Len=33
9	1.625537	172.31.10.74	142.250.194.138	UDP	75	64376 → 443 Len=33
11	1.856012	172.31.10.74	142.250.194.138	UDP	75	64376 → 443 Len=33
13	2.089352	172.31.10.74	142.250.194.138	UDP	75	64376 → 443 Len=33
16	2.320466	172.31.10.74	142.250.194.138	UDP	75	64376 → 443 Len=33
18	2.548838	172.31.10.74	142.250.194.138	UDP	75	64376 → 443 Len=33
21	2.750859	172.31.10.74	54.251.214.91	TCP	66	61928 → 443 [ACK] Seq=1 Ack=441 Win=2041 Len=0 TStamp=3923007324 TSecr=861122707
22	2.985379	172.31.10.74	142.250.194.138	UDP	75	64376 → 443 Len=33
24	3.837194	172.31.10.74	142.250.194.138	UDP	75	64376 → 443 Len=33
27	4.931062	172.31.10.74	172.16.100.205	DNS	74	Standard query 0xfdff A www.google.com
28	4.931240	172.31.10.74	172.16.100.205	DNS	74	Standard query 0xf9ac HTTPS www.google.com
31	4.946276	172.31.10.74	142.250.194.132	QUIC	1292	Initial, DCID=94358cc1d7e6999e, PKN: 1, CRYPTO, PADDING, CRYPTO, CRYPTO, CRYPTO, PING, CRYPTO, PING
32	4.946501	172.31.10.74	142.250.194.132	QUIC	119	0-RTT, DCID=94358cc1d7e6999e
33	4.946779	172.31.10.74	142.250.194.132	QUIC	1288	0-RTT, DCID=94358cc1d7e6999e
34	4.946783	172.31.10.74	142.250.194.132	QUIC	924	0-RTT, DCID=94358cc1d7e6999e
37	5.037379	172.31.10.74	142.250.194.132	QUIC	120	Handshake, DCID=d4358cc1d7e6999e
41	5.039622	172.31.10.74	142.250.194.132	QUIC	75	Protected Payload (K90), DCID=d4358cc1d7e6999e
43	5.057705	172.31.10.74	142.250.194.132	QUIC	75	Protected Payload (K90), DCID=d4358cc1d7e6999e
45	5.095863	172.31.10.74	142.250.194.132	QUIC	75	Protected Payload (K90), DCID=d4358cc1d7e6999e
49	5.154593	172.31.10.74	142.250.194.132	QUIC	77	Protected Payload (K90), DCID=d4358cc1d7e6999e
51	5.181100	172.31.10.74	142.250.194.132	QUIC	75	Protected Payload (K90), DCID=d4358cc1d7e6999e

6) 3-way TCP handshaking

I visited the website www.cricinfo.com and captured packets in wireshark.

No.	#	Time	Source	Destination	Protocol	Length	Info
1	8.000000	172.31.10.74	142.256.266.142	UDP	75	49746 -> 443 Len=33	
2	8.025621	142.256.266.142	172.31.10.74	UDP	68	443 -> 49746 Len=26	
3	8.234272	172.31.10.74	142.256.266.142	UDP	75	49746 -> 443 Len=33	
4	8.261306	142.256.266.142	172.31.10.74	UDP	68	443 -> 49746 Len=26	
5	8.670836	172.31.10.74	142.256.266.142	UDP	75	49746 -> 443 Len=33	
6	8.709840	142.256.266.142	172.31.10.74	UDP	68	443 -> 49746 Len=26	
7	8.158947	172.31.10.74	142.256.266.142	UDP	75	49746 -> 443 Len=33	
8	8.541223	142.256.266.142	172.31.10.74	UDP	68	443 -> 49746 Len=26	
9	8.722586	Cisco_cc:9b:20	Broadcast	ARP	42	Gratuitous ARP for 172.31.17.70 (Reply)	
10	8.7415591	Cisco_cc:9b:20	Broadcast	ARP	42	Gratuitous ARP for 172.31.11.72 (Reply)	
11	8.7512326	Cisco_cc:9b:20	Broadcast	ARP	42	Gratuitous ARP for 172.31.31.77.174 (Reply)	
12	8.842313	172.31.10.74	142.256.194.132	QUIC	1209	Initial, DCID=d00f48723df9724, PKN: 1, PADDING, PING, CRYPTO, PADDING, CRYPTO, PADDING, CRYPTO, C	
13	8.842496	172.31.10.74	142.256.194.132	QUIC	116	0-RTT, DCID=d00f48723df9724	
14	8.842923	172.31.10.74	142.256.194.132	QUIC	1288	0-RTT, DCID=d00f48723df9724	
15	8.842956	172.31.10.74	142.256.194.132	QUIC	885	0-RTT, DCID=d00f48723df9724	
16	8.829253	142.256.194.132	172.31.10.74	QUIC	1292	Initial, DCID=d00f48723df9724, PKN: 1, ACK, PADDING	
17	8.941188	172.31.10.74	142.256.194.132	QUIC	1292	Initial, DCID=d00f48723df9724, PKN: 6, PADDING, PING, PADDING	
18	8.945792	142.256.194.132	172.31.10.74	QUIC	831	Protected Payload (K9P9)	
19	8.945793	142.256.194.132	172.31.10.74	QUIC	217	Protected Payload (K9P9)	
20	8.945794	142.256.194.132	172.31.10.74	QUIC	66	Protected Payload (K9P9)	
21	8.945795	142.256.194.132	172.31.10.74	QUIC	120	Handshake, DCID=d00f48723df9724	
22	8.946179	172.31.10.74	142.256.194.132	QUIC	75	Protected Payload (K9P9), DCID=d00f48723df9724	
23	8.946179	172.31.10.74	142.256.194.132	QUIC	1292	Protected Payload (K9P9)	
24	8.960157	172.31.10.74	142.256.194.132	QUIC	1292	Protected Payload (K9P9), DCID=d00f48723df9724	
25	8.960365	172.31.10.74	142.256.194.132	QUIC	155	Protected Payload (K9P9), DCID=d00f48723df9724	
26	8.964537	142.256.194.132	172.31.10.74	QUIC	162	Protected Payload (K9P9)	
27	8.964653	172.31.10.74	142.256.194.132	QUIC	75	Protected Payload (K9P9), DCID=d00f48723df9724	
28	8.964938	142.256.194.132	172.31.10.74	QUIC	1114	Protected Payload (K9P9)	
29	8.965435	142.256.194.132	172.31.10.74	QUIC	64	Protected Payload (K9P9)	
30	8.965435	142.256.194.132	172.31.10.74	QUIC	79	Protected Payload (K9P9), DCID=d00f48723df9724	
31	8.965459	172.31.10.74	142.256.194.132	QUIC	75	Protected Payload (K9P9), DCID=d00f48723df9724	
32	8.973186	142.256.194.132	172.31.10.74	QUIC	69	Protected Payload (K9P9)	
33	8.115688	172.31.10.74	137.239.91.48	TCP	66	60734 -> 443 [FIN, ACK] Seq=1 Ack=1 Win=0 TStamp=2852751755 TSecr=1248297068	
34	8.115912	172.31.10.74	172.16.106.205	DNS	76	Standard query 0x1e8b A www.cricinfo.com	
35	8.116094	172.31.10.74	172.16.106.205	DNS	76	Standard query 0x267 HTTPS www.cricinfo.com	
36	8.149038	172.31.10.74	142.256.266.142	UDP	75	49746 -> 443 Len=33	
37	8.162276	177.239.91.48	172.31.10.74	TLSv1-	97	Encrypted Alert	
38	8.162277	177.239.91.48	172.31.10.74	TCP	66	443 -> 60734 [FIN, ACK] Seq=32 Ack=2 Win=0 Len=0 TStamp=1248303373 TSecr=2852751755	
39	8.162496	172.31.10.74	137.239.91.48	TCP	54	60734 -> 443 [RST] Seq=2 Ack=2 Win=0 Len=0	
40	8.162665	172.31.10.74	137.239.91.48	TCP	54	60734 -> 443 [RST] Seq=2 Win=0 Len=0	
41	8.174696	142.256.206.142	172.31.10.74	UDP	68	443 -> 49746 Len=26	
42	8.251889	172.31.10.205	172.31.10.74	DNS	433	Standard query response 0x1eb A www.cricinfo.com CNAME origin-hs.espnccricinfo.com CNAME legacy-ll	
43	8.3264253	172.31.10.205	172.31.10.74	DNS	280	Standard query response 0x267 HTTPS www.cricinfo.com CNAME origin-hs.espnccricinfo.com CNAME Len=	

TCP uses the 3-way handshaking protocol to establish a connection between two devices in a network, and after these steps, both devices begin to communicate (transmit data).

The three steps are

- 1) SYN (synchronize)
 - 2) SYN-ACK (synchronize - acknowledge)
 - 3) ACK (acknowledge)

To show the 3-way TCP handshaking we need to use the filter “`tcp.port == 80`”. This filter will show us only the packets on port 80 i.e. HTTP port. The following is the screenshot of the same:

tcp.port == 80								
No.	Time	Source	Destination	Protocol	Length	Info		
44	3.264614	172.31.10.74	18.136.58.21	TCP	78	60744 -> 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=3246945890 TSeср=0 SACK_PERM		
45	3.359967	18.136.58.21	172.31.10.74	TCP	74	80 -> 60744 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1460 SACK_PERM TSval=4227540563 TSeср=324694589...		
46	3.360114	172.31.10.74	18.136.58.21	TCP	66	60744 -> 80 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=3246945986 TSeср=4227540563		
47	3.360188	172.31.10.74	18.136.58.21	HTTP	515	GET / HTTP/1.1		
48	3.363301	18.136.58.21	172.31.10.74	TCP	66	80 -> 60744 [ACK] Seq=1 Ack=450 Win=56576 Len=0 TSval=4227540563 TSeср=3246945986		
49	3.534986	18.136.58.21	172.31.10.74	TCP	66	[TCP Window Update] 80 -> 60744 [ACK] Seq=1 Ack=450 Win=28160 Len=0 TSval=4227540748 TSeср=3246945986		
50	3.654199	18.136.58.21	172.31.10.74	HTTP	628	HTTP/1.1 301 Moved Permanently (text/html)		
51	3.654304	172.31.10.74	18.136.58.21	TCP	66	60744 -> 80 [ACK] Seq=450 Ack=563 Win=131200 Len=0 TSval=3246946280 TSeср=4227540748		

We can clearly see all the 3 ways SYN, SYN-ACK, and ACK in the info section of the above screenshot.

7) Why DNS uses UDP but HTTP uses TCP

UDP is a connectionless transfer protocol. It has no reliability. DNS responses being small in size does not require reliability. Therefore DNS uses UDP stream.

Whereas, TCP is a connection oriented transport protocol. It provides reliability in case of data transfer. HTTP responses being large in size requires reliability in data transfer.

Therefore HTTP uses TCP stream.

8) TCP communication in socket program

I ran the following server and client programs on VS code.

```
server2.py > main
1  import socket
2  import os
3
4  def handle_client(conn, addr):
5      print("Connection from: " + str(addr))
6      while True:
7          data = conn.recv(1024).decode()
8          if not data:
9              break
10         try:
11             result = eval(data)
12             conn.send(str(result).encode())
13         except:
14             conn.send("Invalid Expression".encode())
15     conn.close()
16
17 def main():
18     host = 'localhost'
19     port = 5000
20
21     server_socket = socket.socket()
22     server_socket.bind((host, port))
23     server_socket.listen(5)
24
25     while True:
26         conn, addr = server_socket.accept()
27         pid = os.fork()
28         if pid == 0:
29             server_socket.close()
30             handle_client(conn, addr)
31             exit()
32         else:
33             conn.close()
34
35 if __name__ == '__main__':
36     main()
```

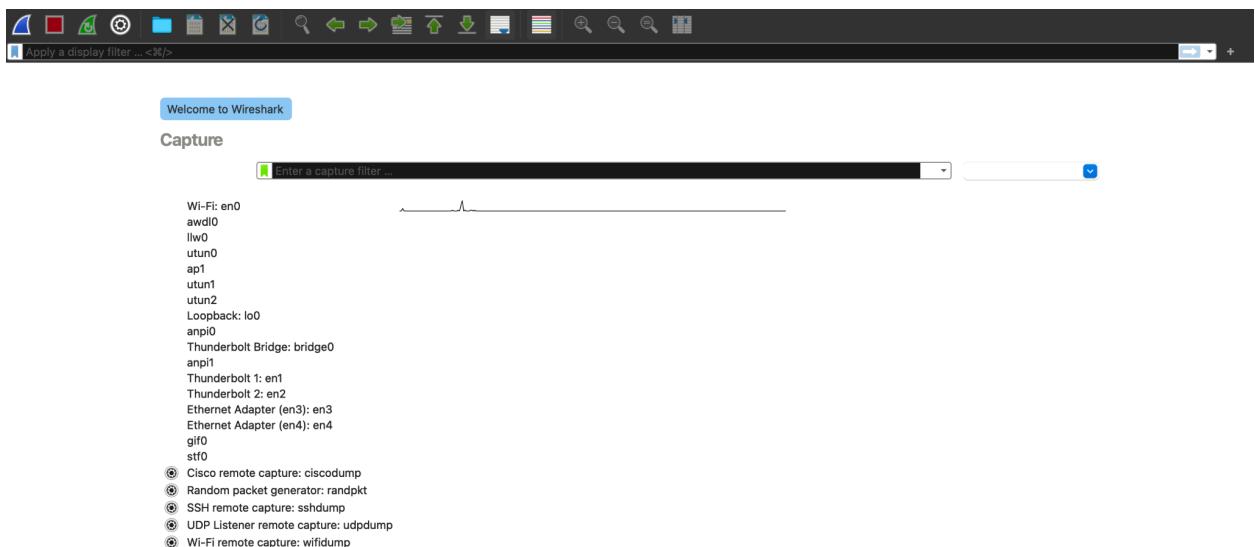
```

client.py > ...
1  import socket
2
3  def main():
4      host = 'localhost'
5      port = 5000
6
7      client_socket = socket.socket()
8      client_socket.connect((host, port))
9
10     message = input("Enter a math expression: ")
11     while message != 'q':
12         client_socket.send(message.encode())
13         data = client_socket.recv(1024).decode()
14         print("Result: " + data)
15         message = input("Enter a math expression: ")
16     client_socket.close()
17
18 if __name__ == '__main__':
19     main()
20

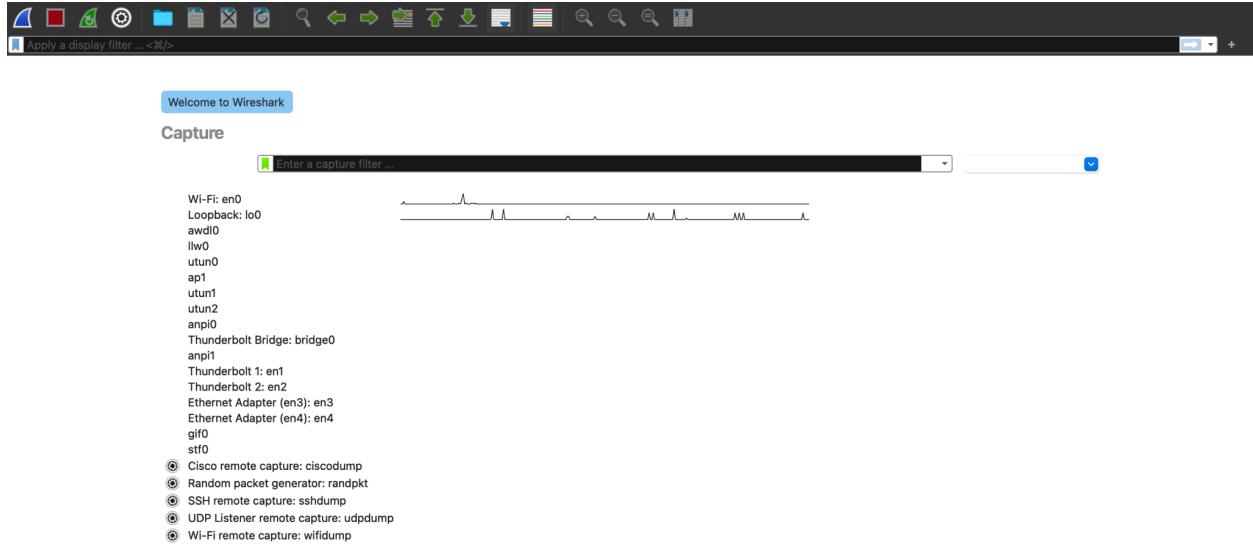
```

It is a simple client-server system where the client chats with a math server.

Before the client-server connection was established the interface of wireshark was looking like this:



Upon establishment of client-server connection the interface looked like :



I clicked on the Loopback. There was no packet capturing happening until this point. Now I supplied the sever following mathematical expressions from the client:

```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE + ⌂ ⌄ ⌁ ⌃ ⌅ ⌆
○ rahulbarodia@Rahuls-MacBook-Air CS_Lab2 % python3 client.py
Enter a math expression: 3+4
Result: 7
Enter a math expression: 2-1
Result: 1
Enter a math expression: (*6
Result: Invalid Expression
Enter a math expression: 8*4
Result: 32
Enter a math expression: 20/4
Result: 5.0
Enter a math expression: 
```

After this, the packet capturing interface at the Wireshark looked like this:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	59	61736 → 5000 [PSH, ACK] Seq=1 Ack=1 Win=6379 Len=3 TSval=719054919 TSecr=1826476326
2	0.000047	127.0.0.1	127.0.0.1	TCP	56	5000 → 61736 [ACK] Seq=1 Ack=4 Win=6379 Len=0 TSval=1826496171 TSecr=719054919
3	0.0000679	127.0.0.1	127.0.0.1	TCP	57	5000 → 61736 [PSH, ACK] Seq=1 Ack=4 Win=6379 Len=1 TSval=1826496171 TSecr=719054919
4	0.000076	127.0.0.1	127.0.0.1	TCP	56	61736 → 5000 [ACK] Seq=4 Ack=2 Win=6379 Len=0 TSval=719054919 TSecr=1826496171
5	2.612153	127.0.0.1	127.0.0.1	TCP	59	61736 → 5000 [PSH, ACK] Seq=4 Ack=2 Win=6379 Len=3 TSval=719057531 TSecr=1826496171
6	2.612199	127.0.0.1	127.0.0.1	TCP	56	5000 → 61736 [ACK] Seq=2 Ack=7 Win=6379 Len=0 TSval=1826498783 TSecr=719057531
7	2.612445	127.0.0.1	127.0.0.1	TCP	57	5000 → 61736 [PSH, ACK] Seq=2 Ack=7 Win=6379 Len=1 TSval=1826498783 TSecr=719057531
8	2.612466	127.0.0.1	127.0.0.1	TCP	56	61736 → 5000 [ACK] Seq=7 Ack=3 Win=6379 Len=0 TSval=719057531 TSecr=1826498783
9	5.574608	127.0.0.1	127.0.0.1	TCP	59	61736 → 5000 [PSH, ACK] Seq=7 Ack=3 Win=6379 Len=3 TSval=719060494 TSecr=1826498783
10	5.574658	127.0.0.1	127.0.0.1	TCP	56	5000 → 61736 [ACK] Seq=3 Ack=10 Win=6379 Len=0 TSval=1826501746 TSecr=719060494
11	5.574909	127.0.0.1	127.0.0.1	TCP	74	5000 → 61736 [PSH, ACK] Seq=3 Ack=10 Win=6379 Len=18 TSval=1826501746 TSecr=719060494
12	5.574934	127.0.0.1	127.0.0.1	TCP	56	61736 → 5000 [ACK] Seq=10 Ack=21 Win=6379 Len=0 TSval=719060494 TSecr=1826501746
13	10.935540	127.0.0.1	127.0.0.1	TCP	59	61736 → 5000 [PSH, ACK] Seq=10 Ack=21 Win=6379 Len=3 TSval=719065855 TSecr=1826501746
14	10.935587	127.0.0.1	127.0.0.1	TCP	56	5000 → 61736 [ACK] Seq=21 Ack=13 Win=6379 Len=0 TSval=1826507107 TSecr=719065855
15	10.935764	127.0.0.1	127.0.0.1	TCP	58	5000 → 61736 [PSH, ACK] Seq=21 Ack=13 Win=6379 Len=2 TSval=1826507107 TSecr=719065855
16	10.935779	127.0.0.1	127.0.0.1	TCP	56	61736 → 5000 [ACK] Seq=13 Ack=23 Win=6379 Len=0 TSval=719065855 TSecr=1826507107
17	17.738678	127.0.0.1	127.0.0.1	TCP	60	61736 → 5000 [PSH, ACK] Seq=13 Ack=23 Win=6379 Len=4 TSval=719072658 TSecr=1826507107
18	17.738724	127.0.0.1	127.0.0.1	TCP	56	5000 → 61736 [ACK] Seq=23 Ack=17 Win=6379 Len=0 TSval=1826513910 TSecr=719072658
19	17.740100	127.0.0.1	127.0.0.1	TCP	59	5000 → 61736 [PSH, ACK] Seq=23 Ack=17 Win=6379 Len=3 TSval=1826513912 TSecr=719072658
20	17.740140	127.0.0.1	127.0.0.1	TCP	56	61736 → 5000 [ACK] Seq=17 Ack=26 Win=6379 Len=0 TSval=719072660 TSecr=1826513912

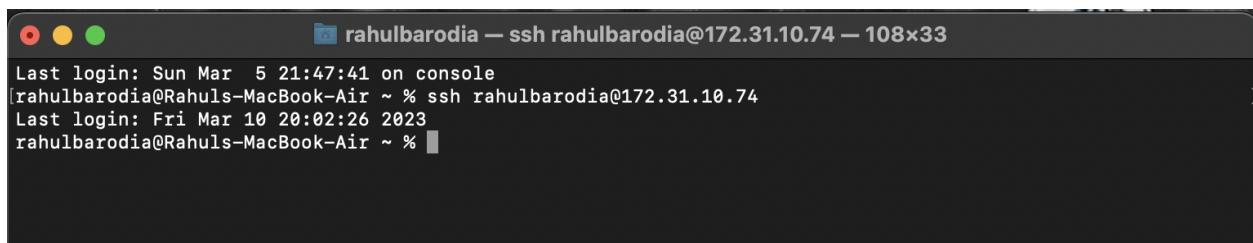
We can clearly observe that TCP communication is happening during client-server communication. The source and destination IP addresses are the same since both the server and client are running on the same machine.

The IP address is 127.0.0.1

We can also observe from the above screenshot that the TCP communication is happening at two different ports. Port no. 5000 and Port no. 61736

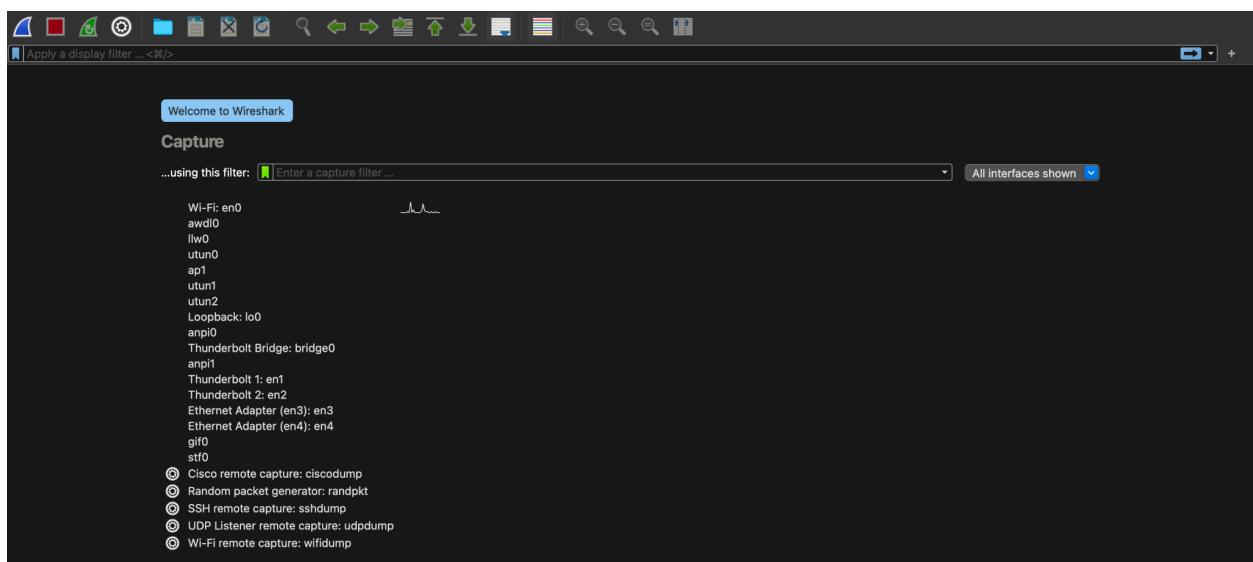
9) Observing packet capturing in SSH connection

I established SSH connection with IITJ home folder as follows:

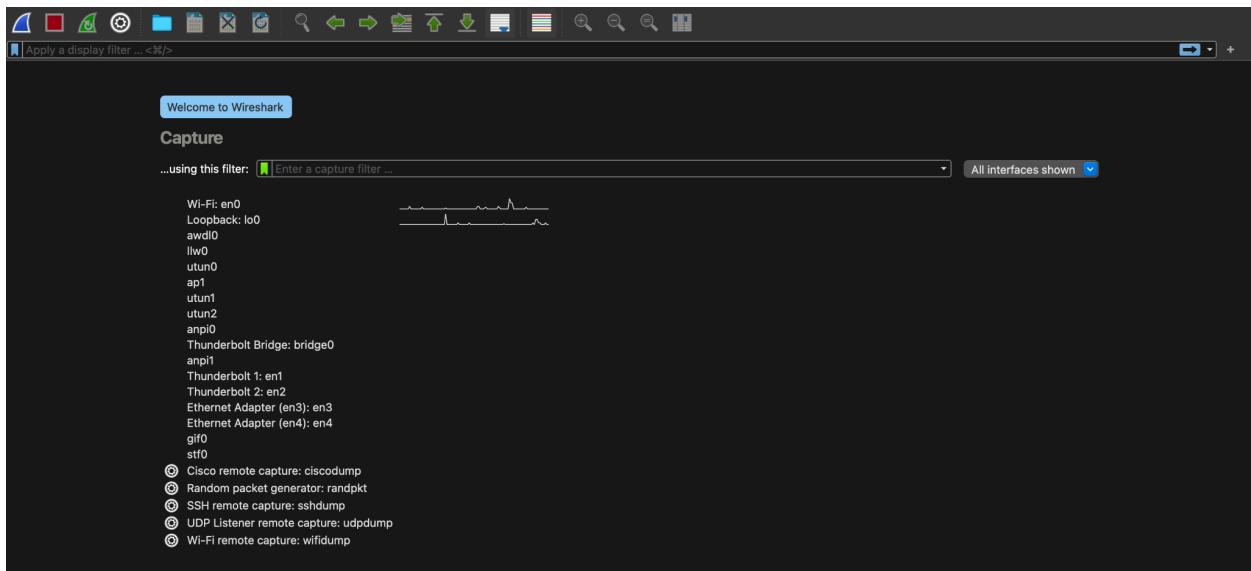


```
rahulbarodia — ssh rahulbarodia@172.31.10.74 — 108x33
Last login: Sun Mar  5 21:47:41 on console
|rahulbarodia@Rahuls-MacBook-Air ~ % ssh rahulbarodia@172.31.10.74
Last login: Fri Mar 10 20:02:26 2023
rahulbarodia@Rahuls-MacBook-Air ~ %
```

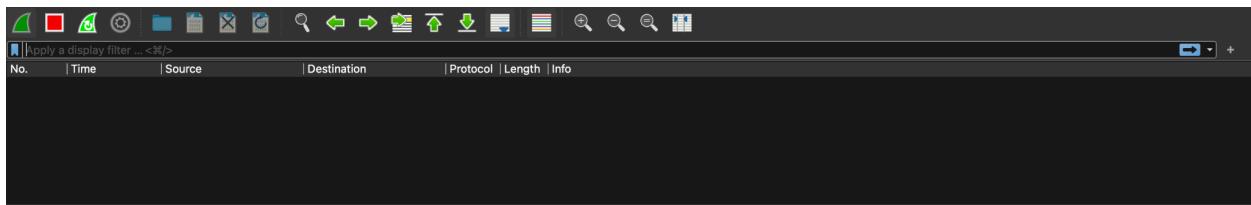
Prior to this the interface of wireshark looked like :



Now it looked like:



I clicked on the Loopback. There was no packet capturing happening until this point.



Then I executed the following command on terminal where SSH connection is established:

```
Last login: Fri Mar 10 20:07:21 on ttys000
[rahulbarodia@Rahuls-MacBook-Air ~ % ssh rahulbarodia@172.31.10.74
Last login: Fri Mar 10 20:07:43 2023
[rahulbarodia@Rahuls-MacBook-Air ~ % ls Desktop
Alexa           Important Docs      Resume.pdf
App_techtut    Lab3                 SSH
B20CS047_A2_CN ML Labs            Screenshots
B20CS047_A2_CN.zip Main Memory    Videos
C++             MyPhoto.jpeg       example.txt
CN Notes        OS Labs            exampleSite
Campus          OS Notes           sample.dat
Codes           PPL_PA            style.txt
DBMS Lab        PRML               theit.txt
DBMS Notes     Pdfs                trial.txt
DD Labs         Photos              Portfolio Website
HTML            rahulbarodia@Rahuls-MacBook-Air ~ %
```

The Wireshark captured the packets and the results looked like:

Wireshark screenshot showing a list of captured network packets. The table has columns: No., Time, Source, Destination, Protocol, Length, and Info. The Info column contains detailed information about each packet, such as 'Client: Encrypted packet (len=36)' or 'Server: Encrypted packet (len=36)'. The sequence numbers (Seq) and acknowledgement numbers (Ack) are also visible in the info column.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.31.10.74	172.31.10.74	SSH	92	Client: Encrypted packet (len=36)
2	0.000080	172.31.10.74	172.31.10.74	TCP	56	22 → 61837 [ACK] Seq=1 Ack=37 Win=6321 Len=0 TSval=2073771685 TSecr=2221899855
3	0.000080	172.31.10.74	172.31.10.74	SSH	92	Server: Encrypted packet (len=36)
4	0.000035	172.31.10.74	172.31.10.74	TCP	56	61837 → 22 [ACK] Seq=37 Ack=37 Win=6323 Len=0 TSval=2221899855 TSecr=2073771685
5	0.517395	172.31.10.74	172.31.10.74	SSH	92	Client: Encrypted packet (len=36)
6	0.517458	172.31.10.74	172.31.10.74	TCP	56	22 → 61837 [ACK] Seq=37 Ack=73 Win=6320 Len=0 TSval=2073772202 TSecr=2221900372
7	0.517790	172.31.10.74	172.31.10.74	SSH	100	Server: Encrypted packet (len=44)
8	0.517817	172.31.10.74	172.31.10.74	TCP	56	61837 → 22 [ACK] Seq=73 Ack=81 Win=6322 Len=0 TSval=2221900372 TSecr=2073772202
9	1.287484	172.31.10.74	172.31.10.74	SSH	92	Client: Encrypted packet (len=36)
10	1.287548	172.31.10.74	172.31.10.74	TCP	56	22 → 61837 [ACK] Seq=81 Ack=109 Win=6320 Len=0 TSval=2073772972 TSecr=2221901142
11	1.287849	172.31.10.74	172.31.10.74	SSH	92	Server: Encrypted packet (len=36)
12	1.287875	172.31.10.74	172.31.10.74	TCP	56	61837 → 22 [ACK] Seq=109 Ack=117 Win=6321 Len=0 TSval=2221901142 TSecr=2073772972
13	1.355222	172.31.10.74	172.31.10.74	SSH	92	Client: Encrypted packet (len=36)
14	1.355274	172.31.10.74	172.31.10.74	TCP	56	22 → 61837 [ACK] Seq=117 Ack=145 Win=6319 Len=0 TSval=2073773040 TSecr=2221901210
15	1.355532	172.31.10.74	172.31.10.74	SSH	100	Server: Encrypted packet (len=44)
16	1.355551	172.31.10.74	172.31.10.74	TCP	56	61837 → 22 [ACK] Seq=145 Ack=161 Win=6321 Len=0 TSval=2221901210 TSecr=2073773040
17	1.612117	172.31.10.74	172.31.10.74	SSH	92	Client: Encrypted packet (len=36)
18	1.612176	172.31.10.74	172.31.10.74	TCP	56	22 → 61837 [ACK] Seq=161 Ack=181 Win=6319 Len=0 TSval=2073773297 TSecr=2221901467
19	1.612461	172.31.10.74	172.31.10.74	SSH	92	Server: Encrypted packet (len=36)
20	1.612486	172.31.10.74	172.31.10.74	TCP	56	61837 → 22 [ACK] Seq=181 Ack=197 Win=6320 Len=0 TSval=2221901467 TSecr=2073773297
21	2.251906	172.31.10.74	172.31.10.74	SSH	92	Client: Encrypted packet (len=36)
22	2.251964	172.31.10.74	172.31.10.74	TCP	56	22 → 61837 [ACK] Seq=197 Ack=217 Win=6318 Len=0 TSval=2073773937 TSecr=2221902107
23	2.252174	172.31.10.74	172.31.10.74	SSH	92	Server: Encrypted packet (len=36)
24	2.252187	172.31.10.74	172.31.10.74	TCP	56	61837 → 22 [ACK] Seq=217 Ack=233 Win=6320 Len=0 TSval=2221902107 TSecr=2073773937
25	2.497685	172.31.10.74	172.31.10.74	SSH	92	Client: Encrypted packet (len=36)
26	2.497819	172.31.10.74	172.31.10.74	TCP	56	22 → 61837 [ACK] Seq=233 Ack=253 Win=6318 Len=0 TSval=2073774183 TSecr=2221902353
27	2.498076	172.31.10.74	172.31.10.74	SSH	92	Server: Encrypted packet (len=36)
28	2.498105	172.31.10.74	172.31.10.74	TCP	56	61837 → 22 [ACK] Seq=253 Ack=269 Win=6319 Len=0 TSval=2221902353 TSecr=2073774183
29	2.641870	172.31.10.74	172.31.10.74	SSH	92	Client: Encrypted packet (len=36)
30	2.642013	172.31.10.74	172.31.10.74	TCP	56	22 → 61837 [ACK] Seq=269 Ack=289 Win=6317 Len=0 TSval=2073774327 TSecr=2221902497
31	2.642240	172.31.10.74	172.31.10.74	SSH	92	Server: Encrypted packet (len=36)
32	2.642268	172.31.10.74	172.31.10.74	TCP	56	61837 → 22 [ACK] Seq=289 Ack=305 Win=6319 Len=0 TSval=2221902497 TSecr=2073774327
33	3.035862	172.31.10.74	172.31.10.74	SSH	92	Client: Encrypted packet (len=36)
34	3.035921	172.31.10.74	172.31.10.74	TCP	56	22 → 61837 [ACK] Seq=305 Ack=325 Win=6316 Len=0 TSval=2073774721 TSecr=2221902891
35	3.036036	172.31.10.74	172.31.10.74	SSH	92	Server: Encrypted packet (len=36)
36	3.036049	172.31.10.74	172.31.10.74	TCP	56	61837 → 22 [ACK] Seq=325 Ack=341 Win=6318 Len=0 TSval=2221902891 TSecr=2073774721

We can analyze this SSH traffic.

Some observations are:

- 1) The source and destination IP addresses are the same because the SSH connection is established between the same machine.
- 2) SSH uses TCP as its transfer protocol.
- 3) The two connected ports have port no. 22 and 61837. SSH typically runs on port no. 22.
- 4) Since SSH connection is encrypted the data transmitted is not visible in the info section. Instead, we can see info such as Client: Encrypted packet (len=36)
- 5) Since it is a TCP connection, the way handshaking is happening. But SYN and SYN-ACK are hidden because the connection is secure. Wireshark can only detect the ACK step which the server sends to the client as the last step before establishing a connection.