

WIRESHARK TOOL

NETWORK ANALYSIS USING WIRESHARK TOOL

INDEX

1.	Different protocols that appear in the protocol column in the unfiltered packet-listing window in Wireshark GUI.
2.	Applying Different Filters in Wireshark to filter network Packets.
3.	Identifying the Internet (IP) address of the URLs visited during the capture and listing the IP address with the site URL.
4.	Calculate total number of captured packets for each protocol.
5.	Find out the IP addresses of the client and server using statistics tool of Wireshark.
6.	Evaluating the total Number of lost packets using Wireshark.
7.	Capturing one TCP 3-way handshake and explaining the process.
8.	TCP packet Analysis and exploring the features in the packet header window. [TCP header and IP header details for the selected packet.]
9.	Exploring the Follow TCP stream feature in Wireshark.
10.	Port Mirroring [SPAN] and Network Analysis using Wireshark.

1. Different protocols that appear in the protocol column in the unfiltered packet-listing window in Wireshark GUI.

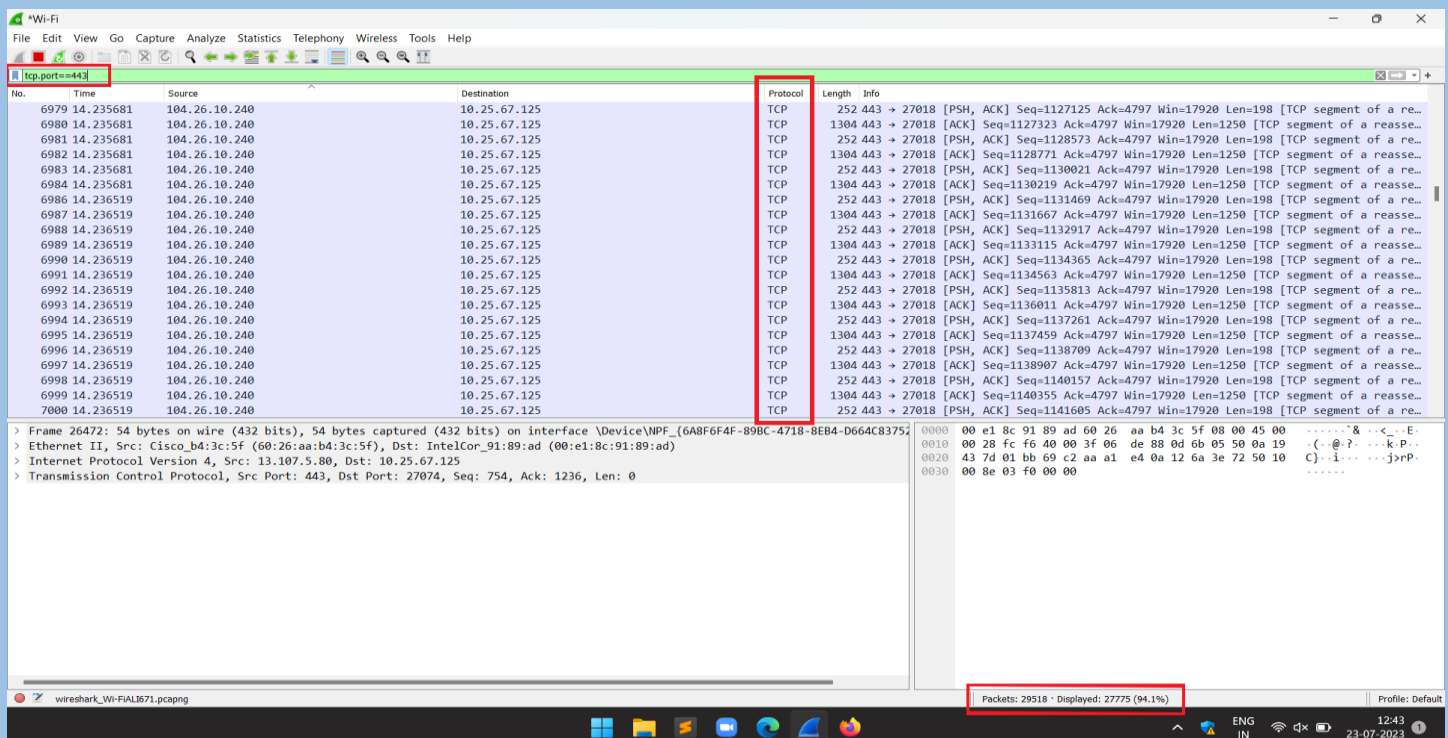
1. Protocols in Networking

- Network protocols are a set of rules outlining how connected devices communicate across a network to exchange information easily and safely.
- Protocols can be Categorized into three categories:
 - ➔ Network Communication Protocols: HTTP, TCP, UDP, FTP etc.
 - ➔ Network Security Protocols: SFTP, HTTPS, SSL etc.
 - ➔ Network Management Protocols: SNMP, ICMP etc.

2. Protocols Observed during Network Analysis using Wireshark Tool

A: TCP [Transmission Control Protocol]

- TCP is a connection-oriented protocol that provides reliable, ordered, and error-checked delivery of data. It uses port numbers to identify the source and destination applications or services.
- Common TCP ports include:
 - Port 80: HTTP (Hypertext Transfer Protocol)
 - Port 443: HTTPS (HTTP Secure)
- Filters used to filter TCP (HTTPS) packets
 1. tcp.port==443 [To get the HTTPS traffic]



2. tcp [To get the packets using TCP protocol]

Wireshark network traffic capture showing TCP packets. The packet list on the left shows a series of TCP packets from 23.3.70.91 to 10.25.67.125. The packet details pane shows the structure of a TCP segment, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane shows the raw hex and ASCII data. A red box highlights the 'tcp' filter in the packet list and the 'Reassembled TCP (4065 bytes)' section in the packet bytes pane.

B: UDP [User Datagram Protocol]

- UDP is a connectionless protocol that does not provide guaranteed delivery or error checking. It is often used for applications that require low latency and do not require reliability. UDP uses port numbers to identify the source and destination applications or services.
- Common UDP ports include:
 - Port 53: DNS (Domain Name System)
 - Port 67/68: DHCP (Dynamic Host Configuration Protocol)
- Filter used to filter UDP packets: `udp`

Wi-Fi capture window showing network traffic. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The main display area is divided into three panes: a packet list on the left, a packet details pane in the middle, and a packet bytes pane on the right. The packet list shows a series of DNS queries and responses. The packet details pane shows the structure of a QUIC packet, including the header, source address, and destination address. The packet bytes pane shows the raw data of the packet, including the header and payload. The status bar at the bottom indicates the current packet number (1292) and the total number of packets (116018).

No.	Time	Source	Destination	Protocol	Length	Info
1134...	1833.069322	10.25.1.1	10.25.67.125	DNS	146	Standard query response 0x858a HTTPS dns.google SOA ns1.zdns.google
1134...	1833.069322	10.25.1.1	10.25.67.125	DNS	182	Standard query response 0x234d A dns.google A 8.8.8.8 A 8.8.4.4
1133...	1817.494291	10.25.1.1	10.25.67.125	DNS	210	Standard query response 0x6928 A teams.events.data.microsoft.com CNAME teams-event...
1123...	1632.383099	10.25.1.1	10.25.67.125	DNS	313	Standard query response 0x9e98 AAAA d2wc28sc48ztgm.cloudfront.net AAAA 2600:9000:2...
1123...	1632.372504	10.25.1.1	10.25.67.125	DNS	183	Standard query response 0x6666 A live.primis.tech CNAME d2wc28sc48ztgm.cloudfront...
1123...	1632.368865	10.25.1.1	10.25.67.125	DNS	153	Standard query response 0xcda4 A d2wc28sc48ztgm.cloudfront.net A 18.66.41.8 A 18.6...
1112...	1450.378604	10.25.1.1	10.25.67.125	DNS	313	Standard query response 0x45a3 AAAA d2wc28sc48ztgm.cloudfront.net AAAA 2600:9000:2...
1111...	1450.309729	10.25.1.1	10.25.67.125	DNS	153	Standard query response 0x5068 A d2wc28sc48ztgm.cloudfront.net A 13.227.254.41 A 1...
1111...	1450.241311	10.25.1.1	10.25.67.125	DNS	183	Standard query response 0xa49d A live.primis.tech CNAME d2wc28sc48ztgm.cloudfront...
1111...	1430.078675	10.25.1.1	10.25.67.125	DNS	157	Standard query response 0x99f2 No such name A wpad.scit.symbisic.soc SOA a.root-se...
1111...	1430.078675	10.25.1.1	10.25.67.125	DNS	157	Standard query response 0xf385 No such name A wpad.scit.symbisic.soc SOA a.root-se...
1109...	1413.328992	10.25.1.1	10.25.67.125	DNS	195	Standard query response 0xe1f0 AAAA youtube-ui.l.google.com AAAA 2404:6800:4009:82...
1109...	1413.326786	10.25.1.1	10.25.67.125	DNS	339	Standard query response 0x2e4b A youtube-ui.l.google.com A 142.250.183.110 A 142.2...
1109...	1413.322212	10.25.1.1	10.25.67.125	DNS	365	Standard query response 0x926c A www.youtube.com CNAME youtube-ui.l.google.com A 1...
1100...	1310.588496	10.25.1.1	10.25.67.125	DNS	141	Standard query response 0x47f4 AAAA e2701.dsca.akamaiedge.net AAAA 2600:1417:3f:6b...
1100...	1310.556443	10.25.1.1	10.25.67.125	DNS	313	Standard query response 0x1b03 AAAA d1elgm1w0d6w0.cloudfront.net AAAA 2600:9000:2...

Frame 10410: 1292 bytes on wire (10336 bits), 1292 bytes captured (10336 bits) on interface \Device\NPF... 8BC-4718-8EB4-D6...
Ethernet II, Src: IntelCor_91:89:ad (00:e1:8c:91:89:ad), Dst: Cisco_b4:3c:5f (60:26:aa:b4:3c:5f)
Internet Protocol Version 4, Src: 10.25.67.125, Dst: 142.250.66.14
0100 = Version: 4
.... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 1278
Identification: 0xa5d1 (42449)
> 010. = Flags: 0x2, Don't fragment
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
Protocol: UDP (17)
Header checksum: 0x317f [validation disabled]
[Header checksum status: Unverified]
Source Address: 10.25.67.125
Destination Address: 142.250.66.14
> User Datagram Protocol, Src Port: 53612, Dst Port: 443
> QUIC IETF

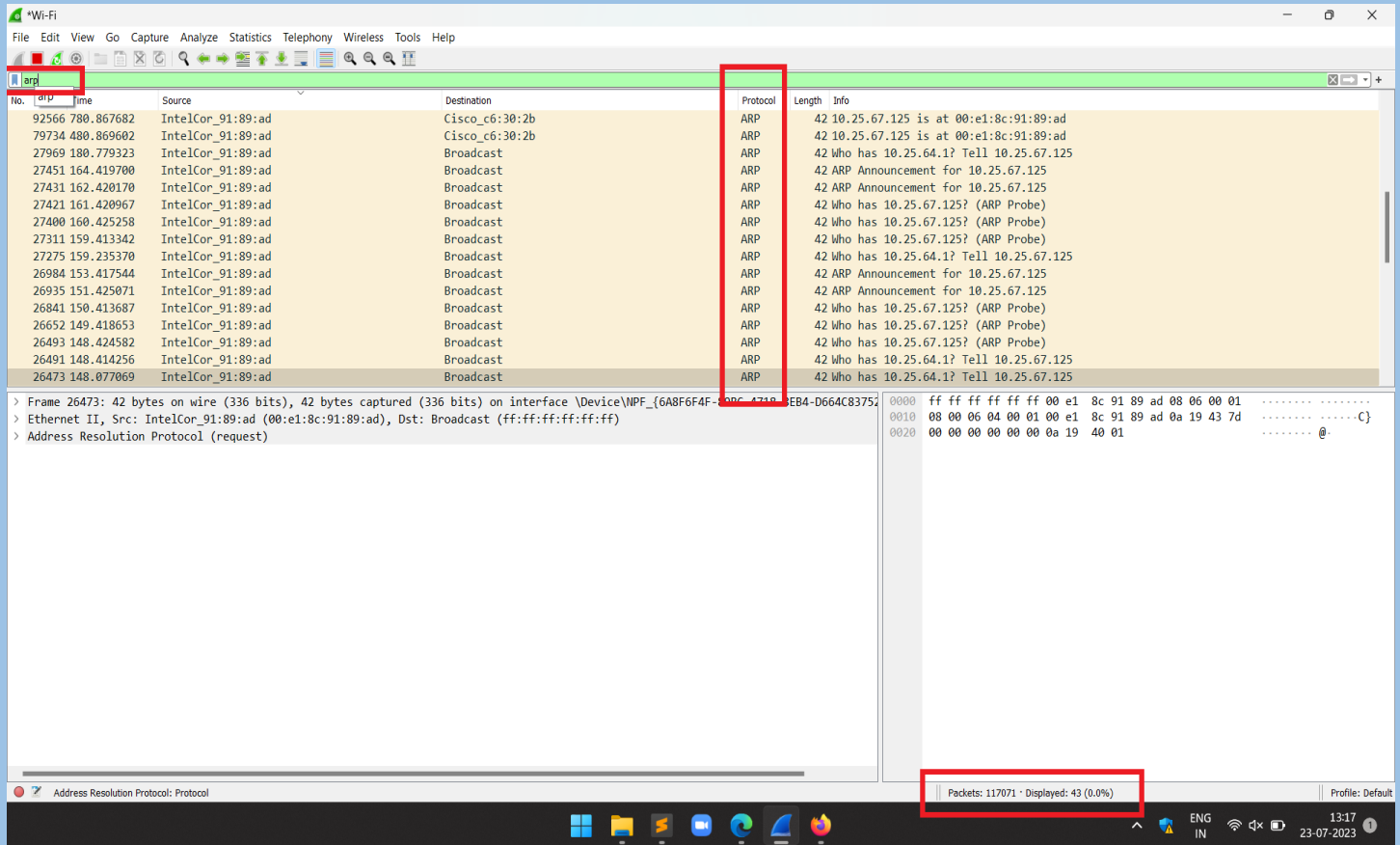
Frame (1292 bytes) Decrypted QUIC (1215 bytes)
Packets: 116018 - Displayed: 2746 (2.4%)

Wi-Fi capture window showing network traffic. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The main display area is divided into three panes: a packet list on the left, a packet details pane in the middle, and a packet bytes pane on the right. The packet list shows a series of DNS queries and responses. The packet details pane shows the structure of a QUIC packet, including the header, source address, and destination address. The packet bytes pane shows the raw data of the packet, including the header and payload. The status bar at the bottom indicates the current packet number (26435) and the total number of packets (29536).

No.	Time	Source	Destination	Protocol	Length	Info
26435	1399.049115	10.25.67.125	142.250.192.14	QUIC	1399	Initial, DCID=da69a95aa0c4f2c7, SCID=1871b5, PKN: 9, CRYPTO
26434	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=da69a95aa0c4f2c7, SCID=1871b5, PKN: 10, PING, PADDING
26433	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=3f62a44088c328047414caaa12451, SCID=9ecc38, PKN: 11, CRYPTO
26432	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=3f62a44088c328047414caaa12451, SCID=9ecc38, PKN: 12, PING, PADDING
26431	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=5b1a6e14f6cc82fe, SCID=67771d, PKN: 11, CRYPTO
26430	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=5b1a6e14f6cc82fe, SCID=67771d, PKN: 12, PING, PADDING
26429	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26428	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26427	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26426	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26425	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26424	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26423	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26422	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26421	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26420	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26419	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26418	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26417	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26416	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26415	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26414	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26413	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26412	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26411	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26410	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26409	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26408	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26407	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26406	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26405	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26404	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26403	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26402	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26401	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26400	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26399	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26398	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26397	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26396	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26395	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26394	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26393	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26392	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26391	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26390	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26389	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26388	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26387	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26386	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26385	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26384	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26383	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26382	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26381	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26380	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26379	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26378	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26377	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26376	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26375	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26374	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26373	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26372	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26371	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26370	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26369	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26368	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26367	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26366	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26365	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26364	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26363	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26362	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, PING, PADDING
26361	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 11, CRYPTO
26360	142.758192	10.25.67.125	142.251.42.74	QUIC	1399	Initial, DCID=bea0d328e5f4e22fa9, SCID=bd538d, PKN: 12, P

C: ARP [Address Resolution Protocol]

- Address Resolution Protocol (ARP) is a protocol or procedure that connects an ever-changing Internet Protocol (IP) address to a fixed physical machine address, also known as a media access control (MAC) address, in a local-area network (LAN).
- Filter Used to filter ARP packets: arp



2. Applying Different Filters in Wireshark to filter network Packets

Packet Filters:

&& -> Both Filters together

|| -> Any of the provided filter

→ IP-Based Filtering:

- ➔ ip.src==<Source-IP> : Filtering packets on the basis of source IP address.
- ➔ ip.dst==<Destination-IP> : Filtering packets on the basis of Destination IP.
- ➔ ip.addr==<IP-address> : Filtering Packets on the basis of IP address (Source or destination IP)
- ➔ !(ip.addr eq <IP>) : To exclude a IP address Ex: (tcp.port==443)&&!(ip.addr eq <IP>)

→ Port-Based Filtering:

```
1. udp.port==<port-number>
```

➔ `udp.srcport==<>`

➔ `udp.dstport==<>`

2. tcp.port==<Port-number>

→ tcp.srcport==<>

➔ tcp.dstport==<>

→ Protocol-Based Filtering:

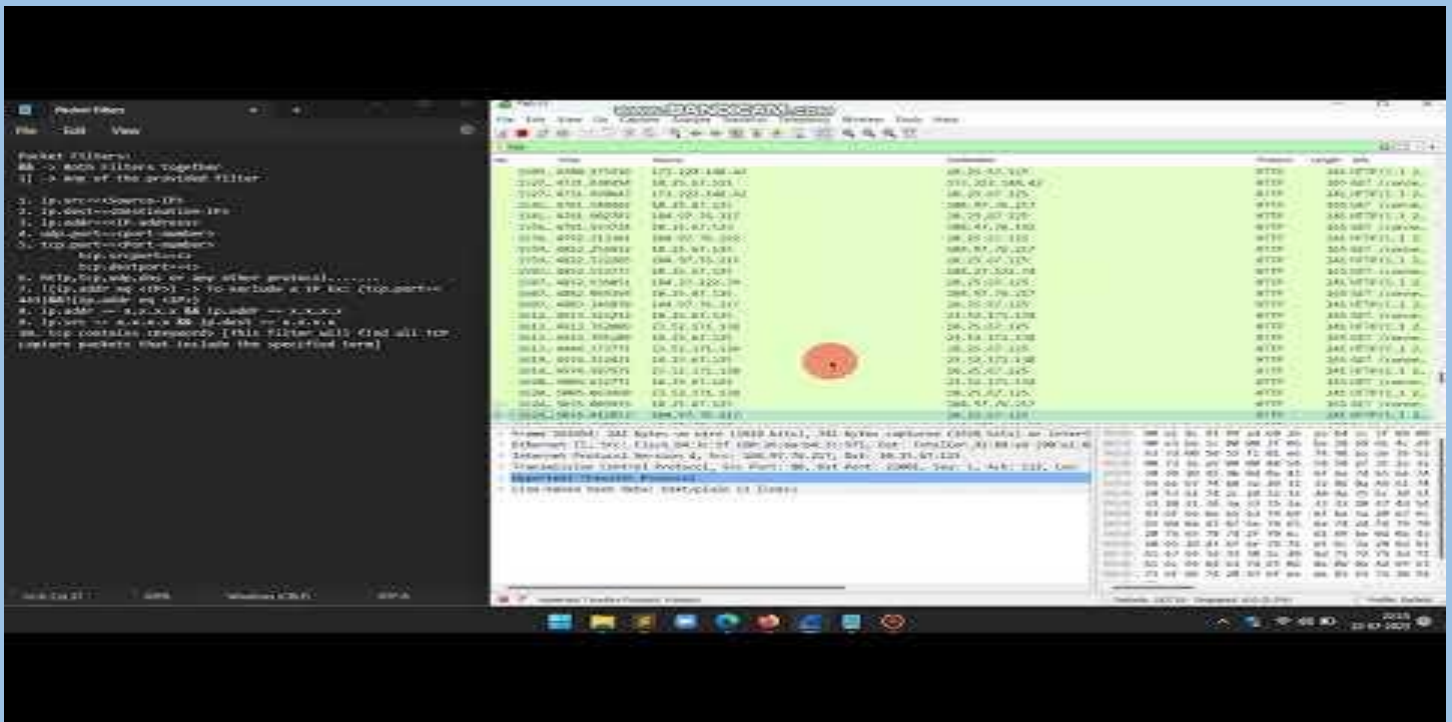
http, tcp, udp, dns, arp or any other protocol.

→ Multiple Filters together:

1. (tcp.port==443)&&! (ip.addr eq <IP>) : Display HTTPS traffic but exclude the traffic from a particular IP address.

2. `ip.addr==<IP>|udp.port==<Port-Number>` : Display Traffic for particular IP address or for Port number which uses UDP protocol (like DHCP, DNS etc..)

Packet-Filtering.mp4 [See video PoC in the attachments provided if Link is Not Working]



<https://youtu.be/0BaGbSHNLIE>

3. Identifying the Internet (IP) address of the URLs visited during the capture and listing the IP address with the site URL.

URL: `http://testphp.vulnweb.com/`

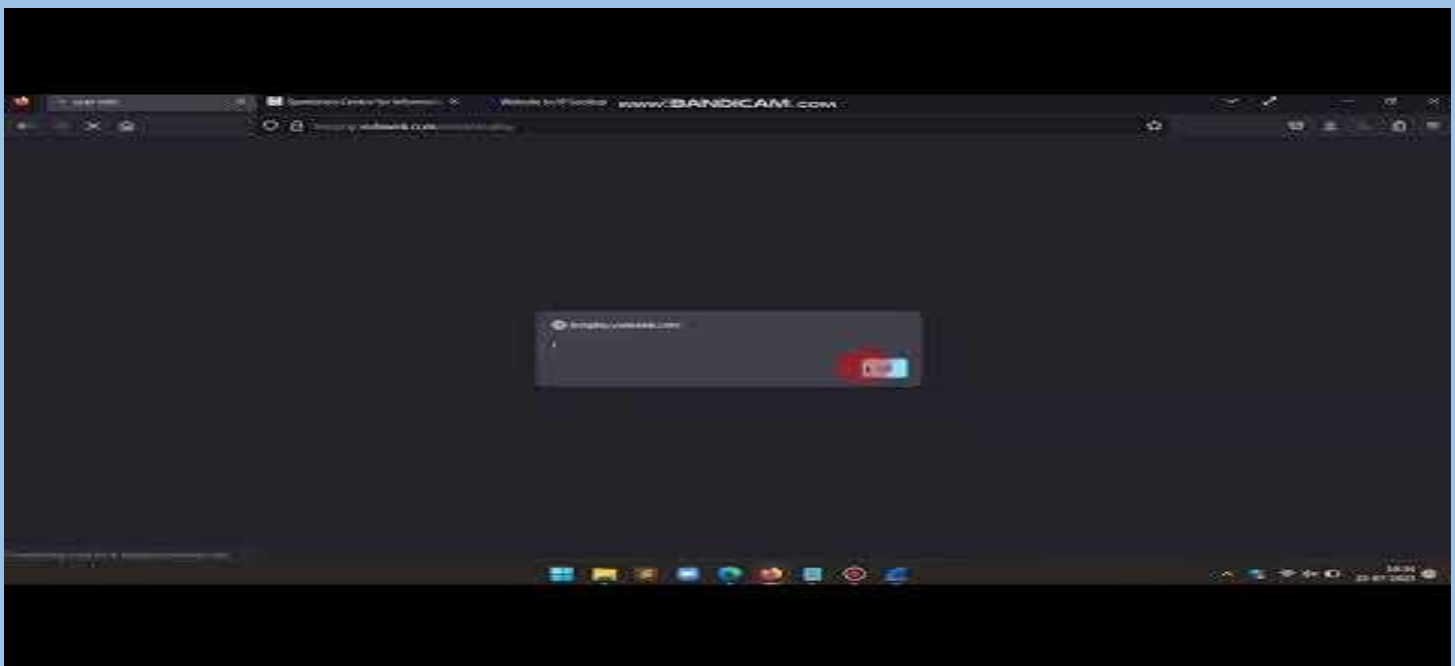
Filter Used: `http contains "http://testphp.vulnweb.com/"`

IP Address: Source-IP (10.25.67.125) and Destination-IP (44.228.249.3)

Additional: Capturing HTTP Traffic: [Credentials are transmitted in plain text]

- ➔ Open Wireshark and start capturing packets by clicking on the blue icon.
- ➔ In the filter tab, add filter "http". [It will sort out and show only HTTP traffic in the Wireshark interface]
- ➔ Go to the website which uses HTTP Instead of HTTPS. [In this case, we are using "`http://testphp.vulnweb.com/login.php`"]
- ➔ Provide login credentials and observe the traffic in Wireshark.
- ➔ Look for Endpoint `"/userinfo.php"` and the HTTP Method will be POST [Because a form is submitted]
- ➔ In that Request, navigate to HTTP Header and look for "cookies". In those cookies, login credentials are transmitted in clear text.
- ➔ Also navigate to HTML Header, under that you can see the login credentials are transmitted in plain text.

HTTP-Traffic.mp4 [See video PoC in the attachments provided if Link is Not Working]



<https://youtu.be/NcnhLK3pnu8>

4. Calculate total number of captured packets for each protocol.[See Point-1 Screenshots]

- TCP: 111818 packets captured out of total packets captured which were 114284. [TCP packets were 97.4% of total packets captured]
- UDP: 2746 Packets captured out of total packets captured which were 116018. [UDP packets were 2.4% of total packets captured]
- ARP: 43 Packets captured out of total packets captured which were 117071.
- HTTP: 179 Packets captured out of total packets captured which were 117837. [HTTP packets were 0.2% of total packets captured]
- DHCP: 6 Packets captured out of total packets captured which were 118863.

5. Find out the IP addresses of the client and server using statistics tool of Wireshark.

The screenshot shows a Wireshark packet capture analysis. The packet list on the left shows a DHCP request from 10.25.67.125 to 255.255.255.255. The packet details pane on the right shows the DHCP request structure. The packet bytes pane on the right shows the raw data.

The screenshot shows a Wireshark packet capture analysis. The packet list on the left shows a DHCP request from 10.25.67.125 to 255.255.255.255. The packet details pane on the right shows the DHCP request structure. The packet bytes pane on the right shows the raw data.

6. Evaluating the total Number of lost packets using Wireshark.

Number of Packets Lost-segment **(While capture is live): 66**

Filter-Used:

1. tcp.analysis.lost_segment: Indicates we've seen a gap in sequence numbers in the capture. Packet loss can lead to duplicate ACKs, which leads to retransmissions.

The image shows a Wireshark packet capture interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for packet capture and analysis. The packet list pane shows a list of captured packets, with the filter 'tcp.analysis.lost_segment' applied. The packet details pane shows the structure of a selected packet, including Ethernet II, Internet Protocol Version 4, and TCP. The packet bytes pane shows the raw data of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
1204...	2794.056622	8.8.8.8	10.25.67.125	TLSv1.3	93	[TCP Previous segment not captured], Application Data
1020...	999.638988	8.8.8.8	10.25.67.125	TLSv1.3	584	[TCP Previous segment not captured], Application Data, Application Data
14522	23.553502	8.8.8.8	10.25.67.125	TLSv1.2	584	[TCP Previous segment not captured], Application Data, Application Data
76919	405.205361	8.8.4.4	10.25.67.125	TLSv1.3	93	[TCP Previous segment not captured], Application Data
30314	325.899350	8.241.166.254	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 80 → 27240 [ACK] Seq=375457 Ack=440 Win=15616 ...
34139	327.123251	8.241.135.126	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 80 → 27241 [ACK] Seq=894179 Ack=440 Win=15616 ...
94744	909.411719	34.160.152.31	10.25.67.125	TLSv1.3	152	[TCP Previous segment not captured], Application Data, Application Data
11947	21.163506	3.6.20.176	10.25.67.125	TLSv1.3	1304	[TCP Previous segment not captured], Continuation Data
1024...	1000.083282	23.3.70.91	10.25.67.125	TLSv1.3	1304	[TCP Previous segment not captured], Continuation Data
1022...	999.791954	23.3.70.91	10.25.67.125	TLSv1.3	1304	[TCP Previous segment not captured], Continuation Data
1016...	999.473375	23.3.70.91	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 443 → 27488 [ACK] Seq=655420 Ack=3539 Win=1715...
1015...	999.462464	23.3.70.91	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 443 → 27488 [ACK] Seq=594542 Ack=3539 Win=1715...
1015...	999.456451	23.3.70.91	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 443 → 27488 [ACK] Seq=554102 Ack=3539 Win=1715...
1013...	999.443097	23.3.70.91	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 443 → 27488 [ACK] Seq=446174 Ack=3539 Win=1715...
3799	10.365137	23.3.70.73	10.25.67.125	TLSv1.3	252	[TCP Previous segment not captured], Continuation Data
99172	993.318521	23.3.70.48	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 443 → 27470 [ACK] Seq=213883 Ack=2719 Win=2841...
98469	992.552230	23.201.47.171	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 443 → 27474 [ACK] Seq=194606 Ack=1429 Win=1740...
5414	11.918854	23.201.47.171	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 443 → 27012 [ACK] Seq=1038729 Ack=1278 Win=171...
1531	8.230223	23.201.47.171	10.25.67.125	TLSv1.3	1304	[TCP Previous segment not captured], Continuation Data
1000	8.148139	23.201.47.171	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 443 → 27003 [ACK] Seq=242406 Ack=1429 Win=1740...
4076	10.534798	23.201.47.137	10.25.67.125	TLSv1.3	390	[TCP Previous segment not captured], Application Data
91119	777.286585	212.115.110.216	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 443 → 27389 [ACK] Seq=541361 Ack=1442 Win=1996...
91110	777.286585	212.115.110.216	10.25.67.125	TCP	1304	[TCP Previous segment not captured] 443 → 27389 [ACK] Seq=528861 Ack=1442 Win=1996...

> Frame 25377: 1304 bytes on wire (10432 bits), 1304 bytes captured (10432 bits) on interface \Device\NPF_{6A8F6F4F-89BC-4718-8EB4-...}

> Ethernet II, Src: Cisco_b4:3c:5f (60:26:aa:b4:3c:5f), Dst: IntelCor_91:89:ad (00:e1:8c:91:89:ad)

> Internet Protocol Version 4, Src: 142.250.67.238, Dst: 10.25.67.125

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 1290

Identification: 0xf825 (63525)

> 000. = Flags: 0x0

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 63

Protocol: TCP (6)

Header Checksum: 0x5e4a [validation disabled]

[Header checksum status: Unverified]

Source Address: 142.250.67.238

0010 05 0a f8 25 00 00 3f 06 5e 4a 8e fa 43 ee 0a 19 ...?.. ^J..C...

0020 43 7d 01 bb 6a 1b 99 9a 7f d0 ea bf 83 77 50 10 C...j... ..WP...

0030 01 f9 9a 75 00 00 fe ea 9f e1 32 4f 08 79 93 98 ...u... ..20..y...

0040 ff 09 ef c5 50 f4 21 95 69 6e ae f6 91 63 d5 8f ...P..I.. in...c...

0050 fd a1 69 aa b4 72 b2 9b 9a e3 8c f4 9e 65 77 6b ...i...r... ..ewk...

0060 eb d0 ad 33 7b 5d 5f 8d cf 07 2c c9 09 62 aa 90 ...3{]... ..b...

0070 98 97 4a 22 ee a2 8e c3 e5 ff b0 ce 53 1d fd c7 ...j".....S...

0080 a2 66 0b 41 69 41 05 96 82 ea 66 2e ce f0 98 d0 ...f AiA... ..f.....

0090 0e 4e de 59 8f 47 85 70 e9 ea f7 e4 f5 a7 59 f2 ...N..Y..G..p.....Y...

00a0 50 ca c8 61 5b 9e 04 64 fd 22 0e 99 d0 99 71 26 P...a[...d..."...q&

00b0 51 3f cd bd 3e 39 54 fa 49 04 02 e5 6b da 58 fb Q?...>9T... I...k..X...

00c0 17 6a e6 56 74 22 fd c9 19 f2 51 47 b2 a2 e0 60 ...j..Vt"... ..QG....

00d0 2f fd 9b 64 9e 94 80 f6 1b 57 ae 31 86 c6 c3 11 /...d.... ..W..1....

00e0 9b 0b c6 18 6d b5 b6 64 73 f7 a2 d2 e4 47 da 64 ...m...d s....G..d

00f0 cc c2 ef 9f 7d ee 4b 99 80 e3 33 9f 5f 08 5e ac ...K... ..3...^...

0100 a5 39 7f 59 d9 bd 60 a5 94 15 88 9f ba c8 cd a7 ...9..Y.... ..

0110 92 91 cd 34 ed 1c 17 22 94 60 cb 96 f5 49 96 48 ...4..."... ..I..H

Previous segment(s) not captured (common at capture start): Label

Packets: 131782 · Displayed: 66 (0.1%)

Profile: Default

13:36 23-07-2023

2. tcp.analysis.retransmission: Displays all retransmissions in the capture.

Wireshark - tcp.analysis.retransmission

No.	Time	Source	Destination	Protocol	Length	Info
28051	186.470012	8.8.4.4	10.25.67.125	TCP	66	[TCP Retransmission] 443 → 27199 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1250 SACK_PERM WS=128
27863	170.402179	8.8.4.4	10.25.67.125	TCP	66	[TCP Retransmission] 443 → 27199 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1250 SACK_PERM WS=128
27430	162.404422	8.8.4.4	10.25.67.125	TCP	66	[TCP Retransmission] 443 → 27199 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1250 SACK_PERM WS=128
27240	158.435575	8.8.4.4	10.25.67.125	TCP	66	[TCP Retransmission] 443 → 27199 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1250 SACK_PERM WS=128
27214	156.399618	8.8.4.4	10.25.67.125	TCP	66	[TCP Retransmission] 443 → 27199 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1250 SACK_PERM WS=128
102642	1002.296606	66.225.223.31	10.25.67.125	TCP	85	[TCP Retransmission] 443 → 27478 [FIN, PSH, ACK] Seq=3793 Ack=1648 Win=17792 Len=31
11961	21.168732	3.6.20.176	10.25.67.125	TCP	252	[TCP Retransmission] 443 → 27070 [PSH, ACK] Seq=6126 Ack=1638 Win=17792 Len=198
11960	21.168732	3.6.20.176	10.25.67.125	TCP	1304	[TCP Retransmission] 443 → 27070 [ACK] Seq=4876 Ack=1638 Win=17792 Len=1250
11959	21.168732	3.6.20.176	10.25.67.125	TCP	252	[TCP Retransmission] 443 → 27070 [PSH, ACK] Seq=4678 Ack=1638 Win=17792 Len=198
11958	21.168732	3.6.20.176	10.25.67.125	TCP	1304	[TCP Retransmission] 443 → 27070 [ACK] Seq=3428 Ack=1638 Win=17792 Len=1250
11957	21.168732	3.6.20.176	10.25.67.125	TCP	252	[TCP Retransmission] 443 → 27070 [PSH, ACK] Seq=3230 Ack=1638 Win=17792 Len=198
3822	10.368393	23.3.70.73	10.25.67.125	TCP	1304	[TCP Spurious Retransmission] 443 → 26995 [ACK] Seq=12019 Ack=1921 Win=19712 Len=1250
3821	10.368393	23.3.70.73	10.25.67.125	TCP	252	[TCP Spurious Retransmission] 443 → 26995 [PSH, ACK] Seq=11821 Ack=1921 Win=19712 Len=198
3820	10.368393	23.3.70.73	10.25.67.125	TCP	1304	[TCP Spurious Retransmission] 443 → 26995 [ACK] Seq=10571 Ack=1921 Win=19712 Len=1250
3819	10.368393	23.3.70.73	10.25.67.125	TCP	252	[TCP Spurious Retransmission] 443 → 26995 [PSH, ACK] Seq=10373 Ack=1921 Win=19712 Len=198
3818	10.368393	23.3.70.73	10.25.67.125	TCP	1304	[TCP Spurious Retransmission] 443 → 26995 [ACK] Seq=9123 Ack=1921 Win=19712 Len=1250
3817	10.368393	23.3.70.73	10.25.67.125	TCP	827	[TCP Spurious Retransmission] 443 → 26995 [PSH, ACK] Seq=8350 Ack=1921 Win=19712 Len=773
1030	8.150522	23.201.47.171	10.25.67.125	TCP	264	[TCP Spurious Retransmission] 443 → 27003 [PSH, ACK] Seq=242196 Ack=1429 Win=17408 Len=210
1029	8.150522	23.201.47.171	10.25.67.125	TCP	1304	[TCP Spurious Retransmission] 443 → 27003 [ACK] Seq=237196 Ack=1429 Win=17408 Len=1250
1028	8.150522	23.201.47.171	10.25.67.125	TCP	1304	[TCP Spurious Retransmission] 443 → 27003 [ACK] Seq=235946 Ack=1429 Win=17408 Len=1250
4080	10.536115	23.201.47.137	10.25.67.125	TLSv1.3	227	[TCP Fast Retransmission] , Application Data, Application Data
91153	777.290420	122.115.110...	10.25.67.125	TLSv1.2	1304	[TCP Fast Retransmission] , Application Data
26450	144.291280	204.79.197.2	10.25.67.125	TCP	54	[TCP Retransmission] 443 → 27042 [FIN, ACK] Seq=7184 Ack=1588 Win=19968 Len=0

Frame 28051: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{6A8F6F4F-89BC-4718-8EB4-D664C83} over Ethernet II, Src: Cisco_b4:3c:5f (60:26:aa:b4:3c:5f), Dst: IntelCor_91:89:ad (00:e1:8c:91:89:ad)

Internet Protocol Version 4, Src: 8.8.4.4, Dst: 10.25.67.125

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 52

Identification: 0xf435 (64821)

0000 = Flags: 0x0

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 63

Protocol: TCP (6)

Header Checksum: 0x24ed [validation disabled]

[Header checksum status: Unverified]

Source Address: 8.8.4.4

Packets: 134266 · Displayed: 292 (0.2%)

Profile: Default

Finally, with **Wireshark capture Stop (after have captured all you need/want)**, then go to **Statistics -> Capture File Properties**. There you can see the **Dropped packets (Under Interface Heading)** which is **0.0%** in our case.

Wireshark - Capture File Properties - Wi-Fi

Details

File

Name: C:\Users\SHAILE-1\AppData\Local\Temp\wireshark_Wi-Fi\1671.pcapng

Length: 110 MB

Hash (SHA256): d220bdc5cd561e6ce2d38bf3baafe715e1570fb58478c732f583f11e69f5e

Hash (RIPEMD160): 4bc470599016843570844ee16dc5cd9ee3c5d83

Hash (SHA1): 5104a9866d494f8cc742b0424d6855bd03e414

Format: Wireshark/... - pcapng

Encapsulation: Ethernet

Time

First packet: 2023-07-23 12:38:23

Last packet: 2023-07-23 13:43:20

Elapsed: 01:04:57

Capture

Hardware: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (with SSE4.2)

OS: 64-bit Windows (22H2), build 22621

Application: Dumpcap (Wireshark) 4.0.7 (v4.0.7-0-g0ad1823cc090)

Interfaces

Interface: Wi-Fi

Dropped packets: 0 (0.0%)

Capture filter: none

Link type: Ethernet

Packet size limit (snaplen): 262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	137412	294 (0.2%)	---
Time span, s	3897.027	3858.546	---
Average pps	35.3	0.1	---
Average packet size, B	775	262	---
Bytes	106435516	76908 (0.1%)	0
Average bytes/s	27 k	19	---
Average bits/s	218 k	159	---

Capture file comments

Refresh

Save Comments Close Copy To Clipboard Help

Profile: Default

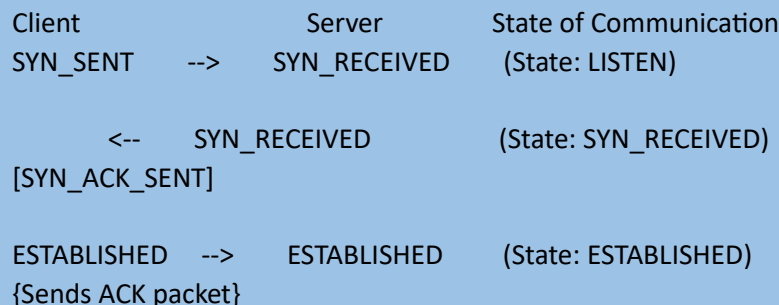
7. Capturing one TCP 3-way handshake and explaining the process.

TCP 3 WAY HANDSHAKE:

1. The sender initiates the communication by sending a SYN packet, expressing their interest in connecting.
2. The recipient responds with a SYN/ACK packet that expresses their interest as well and acknowledges receipt of the sender's SYN packet.
3. Finally, the sender sends an ACK packet to acknowledge the recipient's SYN/ACK.

- TCP (Transmission Control Protocol) uses a three-way handshake to establish a reliable connection between two devices.

- The 3-way handshake process:



1. SYN_SENT:

- The client sends a SYN packet to the server, indicating that it wants to establish a connection.
- The SYN packet includes a random sequence number that is used to identify packets in the connection.
- The client enters the SYN_SENT state, waiting for a response from the server.

2. SYN_RECEIVED:

- The server receives the SYN packet from the client and responds with a SYN-ACK packet.
- The SYN-ACK packet includes a random sequence number and an acknowledgment number that is equal to the client's sequence number plus one.
- The server enters the SYN_RECEIVED state, waiting for the final ACK packet from the client.

3. ESTABLISHED:

- The client receives the SYN-ACK packet from the server and sends an ACK packet back to the server.
- The ACK packet includes the acknowledgment number that was sent in the server's SYN-ACK packet.

Once the server receives the ACK packet, the connection is established, and both devices enter the ESTABLISHED state. They are now ready to exchange data over the connection.

8. TCP packet Analysis and exploring the features in the packet header window. [TCP header and IP header details for the selected packet.]

1. TCP Header

-> Source and Destination Ports

Source port: this is a 16-bit field that specifies the port number of the sender.

Destination port: this is a 16-bit field that specifies the port number of the receiver.

-> Understanding TCP 3 Way handshake by analyzing TCP header of the TCP packets.

TCP utilizes a number of flags, in its header to control the state of a connection.

1. Select packet [SYN] in Wireshark and expand the TCP layer analysis in the middle pane, and further expand the "Flags" field within the TCP header. Here we can see all of the TCP flags broken down. Note that the SYN flag is on (set to 1).

The image shows a Wireshark packet capture window titled "Wi-Fi". The packet list on the left shows several packets, with packet 4302 (a SYN packet) selected and highlighted in red. The packet details pane on the right shows the expanded TCP header for packet 4302. The "Flags" field is expanded, showing the following values:

- Source Port: 60278
- Destination Port: 80
- Sequence Number: 0 (relative sequence number)
- Sequence Number (raw): 2346227018
- Next Sequence Number: 1 (relative sequence number)
- Acknowledgment Number: 0
- Acknowledgment number (raw): 0
- Header Length: 32 bytes (8)
- Flags: 0x002 (SYN)
- Reserved: Not set
- Accurate ECH: Not set
- Congestion Window Reduced: Not set
- ECH-Echo: Not set
- Urgent: Not set
- Acknowledgment: Not set
- Push: Not set
- Reset: Not set
- SYN: Set
- Fin: Not set

The packet bytes pane on the right shows the raw data of the packet, with the first 32 bytes (the TCP header) highlighted in blue. The status bar at the bottom indicates "Packets: 123372 - Displayed: 57 (0.0%)" and "Profile: Default".

2. Now do the same for packet [SYN, ACK]. Notice that it has two flags set: ACK to acknowledge the receipt of the client's SYN packet, and SYN to indicate that the server also wishes to establish a TCP connection.

Wireshark packet capture showing a SYN, ACK packet (packet 4304) from the server to the client. The packet details show the ACK flag set to 1 and the SYN flag set to 1. The packet bytes show the TCP header and the data payload.

Packet 4304: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{6ABF6F4F-89BC-4718-8EB4-D664C837}

Ethernet II, Src: Cisco_b4:3c:5f (60:26:aa:b4:3c:5f), Dst: IntelCor_91:89:ad (00:e1:8c:91:89:ad)

Internet Protocol Version 4, Src: 3.220.135.75, Dst: 10.25.66.55

Transmission Control Protocol, Src Port: 80, Dst Port: 60278, Seq: 0, Ack: 1, Len: 0

Source Port: 80
Destination Port: 60278
[Stream index: 77]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]

Sequence Number: 0 (relative sequence number)
Sequence number (raw): 234604289
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 2346227019
1000 = Header Length: 32 bytes (8)

Flags: 0x012 (SYN, ACK)

000. = Reserved: Not set
...0 = Accurate ECN: Not set
...0 = Congestion Window Reduced: Not set
...0 = ECN-Echo: Not set
...0 = Urgent: Not set
...1 = Acknowledgment: Set
...0 = Push: Not set
...0 = Reset: Not set
...1 = Syn: Set
...0 = Fin: Not set

Sequence Number (tcp.seq), 4 bytes

Packets: 123400 - Displayed: 57 (0.0%)

3. Packet [ACK], from the client, has only the ACK flag set. These three packets complete the initial TCP three-way handshake.

Wireshark packet capture showing an ACK packet (packet 4305) from the client to the server. The packet details show the ACK flag set to 1. The packet bytes show the TCP header and the data payload.

Packet 4305: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{6ABF6F4F-89BC-4718-8EB4-D664C837}

Ethernet II, Src: IntelCor_91:89:ad (00:e1:8c:91:89:ad), Dst: Cisco_b4:3c:5f (60:26:aa:b4:3c:5f)

Internet Protocol Version 4, Src: 10.25.66.55, Dst: 3.220.135.75

Transmission Control Protocol, Src Port: 60278, Dst Port: 80, Seq: 1, Ack: 1, Len: 0

Source Port: 60278
Destination Port: 80
[Stream index: 77]
[Conversation completeness: Complete, WITH_DATA (31)]
[TCP Segment Len: 0]

Sequence Number: 1 (relative sequence number)
Sequence number (raw): 2346227019
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 2564604286
0101 = Header Length: 20 bytes (5)

Flags: 0x010 (ACK)

000. = Reserved: Not set
...0 = Accurate ECN: Not set
...0 = Congestion Window Reduced: Not set
...0 = ECN-Echo: Not set
...0 = Urgent: Not set
...1 = Acknowledgment: Set
...0 = Push: Not set
...0 = Reset: Not set
...0 = Syn: Not set
...0 = Fin: Not set
[TCP Flags:A....]
Window: 512
[Calculated window size: 131072]
[Outgoing sequence number: 131072]

Sequence Number (tcp.seq), 4 bytes

Packets: 123508 - Displayed: 57 (0.0%)

-> Sequence and Acknowledgment Numbers

The client on either side of a TCP session maintains a 32-bit sequence number it uses to keep track of how much data it has sent. This sequence number is included on each transmitted packet, and acknowledged by the opposite host as an acknowledgement number to inform the sending host that the transmitted data was received successfully.

When a host initiates a TCP session, its initial sequence number is effectively random; it may be any value between 0 and 4,294,967,295, inclusive. However, protocol analyzers like Wireshark will typically display relative sequence and acknowledgement numbers in place of the actual values. These numbers are relative to the initial sequence number of that stream. This is handy, as it is much easier to keep track of relatively small, predictable numbers rather than the actual numbers sent on the wire.

For example, the initial relative sequence number shown in packet [SYN] is 0.

2. IP Header

- Source Address (the IP address of the original sender of the packet)
- Destination Address (the IP address of the final destination of the packet)

Wireshark packet capture showing a TCP SYN packet. The packet list shows packet 24333 from 142.250.183.3 to 10.25.67.125. The packet details pane shows the IP header with Source Address 142.250.183.3 and Destination Address 10.25.67.125 highlighted in red. The packet bytes pane shows the raw data.

No.	Time	Source	Destination	Protocol	Length	Info
24348	122.171572	142.250.183.3	10.25.67.125	TCP	54	443 → 27171 [ACK] Seq=85549 Ack=7484 Win=19968 Len=0
24345	122.170847	142.250.183.3	10.25.67.125	TCP	54	443 → 27171 [ACK] Seq=85549 Ack=7290 Win=19968 Len=0
24341	122.170059	142.250.183.3	10.25.67.125	TCP	54	443 → 27171 [ACK] Seq=85549 Ack=7103 Win=19968 Len=0
24340	122.170059	142.250.183.3	10.25.67.125	TCP	54	443 → 27171 [ACK] Seq=85549 Ack=6913 Win=19968 Len=0
24339	122.170059	142.250.183.3	10.25.67.125	TCP	54	443 → 27171 [ACK] Seq=85549 Ack=6626 Win=19968 Len=0
24337	122.169797	142.250.183.3	10.25.67.125	TCP	54	443 → 27171 [ACK] Seq=85549 Ack=6344 Win=19968 Len=0
24335	122.169797	142.250.183.3	10.25.67.125	TCP	54	443 → 27171 [ACK] Seq=85549 Ack=6142 Win=19968 Len=0
24334	122.169797	142.250.183.3	10.25.67.125	TCP	54	443 → 27171 [ACK] Seq=85549 Ack=5947 Win=19968 Len=0
24333	122.169797	142.250.183.3	10.25.67.125	TCP	54	443 → 27171 [ACK] Seq=85549 Ack=5761 Win=19968 Len=0
24326	122.168157	142.250.183.3	10.25.67.125	TCP	54	443 → 27171 [ACK] Seq=85549 Ack=5478 Win=19968 Len=0
24317	122.166747	142.250.183.3	10.25.67.125	TCP	54	443 → 27171 [ACK] Seq=85549 Ack=5179 Win=19968 Len=0

Frame 24333: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{6A8F6F4F-89BC-4718-8EB4-D664C83752} Ethernet II, Src: Cisco b4:3c:5f (60:26:aa:b4:3c:5f), Dst: IntelCor 91:89:ad (00:e1:8c:91:89:ad)

Internet Protocol Version 4, Src: 142.250.183.3, Dst: 10.25.67.125

- 0100 = Version: 4
- 0101 = Header Length: 20 bytes (5)
- > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
- Total Length: 40
- Identification: 0xf4b7 (62647)
- > 000. = Flags: 0x0
- ...0 0000 0000 0000 = Fragment Offset: 0
- Time to Live: 63
- Protocol: TCP (6)
- Header Checksum: 0xf384 [validation disabled]
- [Header checksum status: Unverified]
- Source Address: 142.250.183.3
- Destination Address: 10.25.67.125

Transmission Control Protocol, Src Port: 443, Dst Port: 27171, Seq: 85549, Ack: 5761, Len: 0

9. Exploring the Follow TCP stream feature in Wireshark

TCP-STREAM: [To see the data from a TCP stream in the way that the application layer sees it]

Simply select a TCP packet in the packet list of the stream/connection you are interested in and then select the Follow TCP Stream menu item from the Wireshark Tools menu (or use the context menu in the packet list).

Wireshark will set an appropriate display filter and pop up a dialog box with all the data from the TCP stream laid out in order

The screenshot shows the Wireshark interface with the packet list pane displaying a list of packets. The packet details pane shows the selected packet's structure, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane shows the raw data of the packet. A context menu is open over the selected packet, with the 'Follow' option highlighted. The 'Follow' submenu is also visible, showing options like TCP Stream, UDP Stream, etc.

The screenshot shows the Wireshark interface with the 'Follow TCP Stream' dialog box open. The dialog box displays the selected stream (TCP Stream 77) and the data from the stream, including the HTTP request and response. The 'Follow' submenu is also visible, showing options like TCP Stream, UDP Stream, etc.

Key-Points:

1. The stream content is displayed in the same sequence as it appeared on the network. HTTP Request and then HTTP Response.
2. Traffic from the client to the server is colored red, while traffic from the server to the client is colored blue. [These colors can be changed by opening Edit → Preferences and under Appearance → Font and Colors, selecting different colors for the Sample "Follow Stream" client text and Sample "Follow Stream" server text options.]

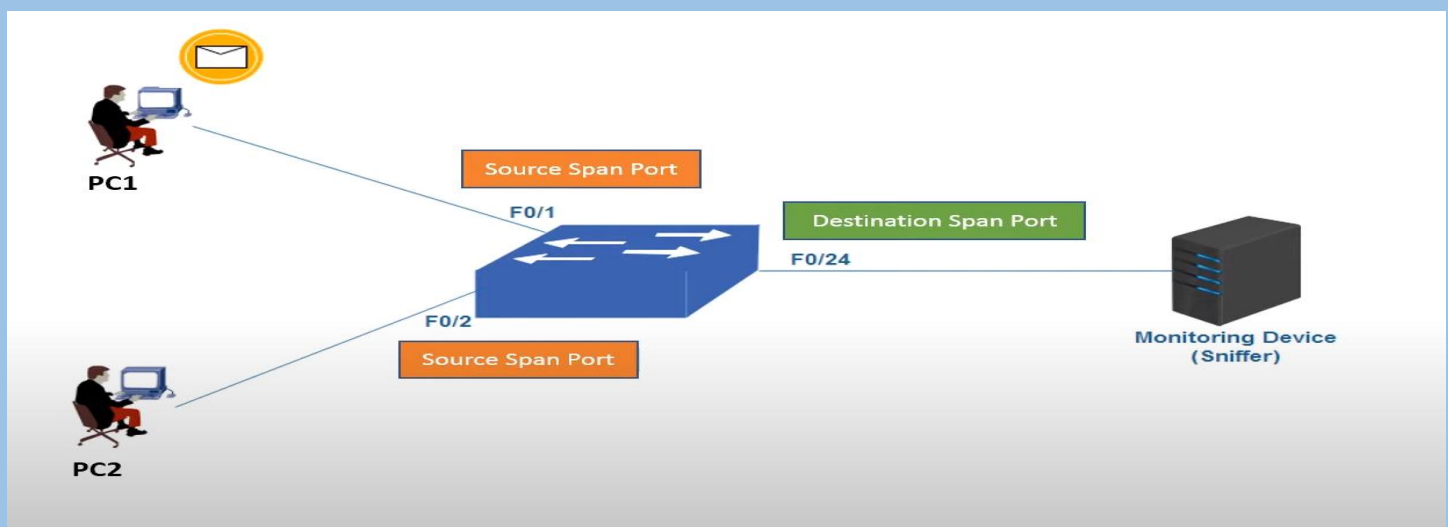
10. Port Mirroring [SPAN Port] and Network Analysis using Wireshark

→ Port Mirroring:

- When port mirroring is enabled, the TOR [Top-of-Rack] switch sends a copy of the network packet from the mirrored ports to the monitor port. This feature is typically used for monitoring and intrusion detection.
- Port mirroring is used on a network switch to send a copy of network packets seen on one switch port (or an entire VLAN) to a network monitoring connection on another switch port.

→ Switch with SPAN [Switched Port Analyzer] Port:

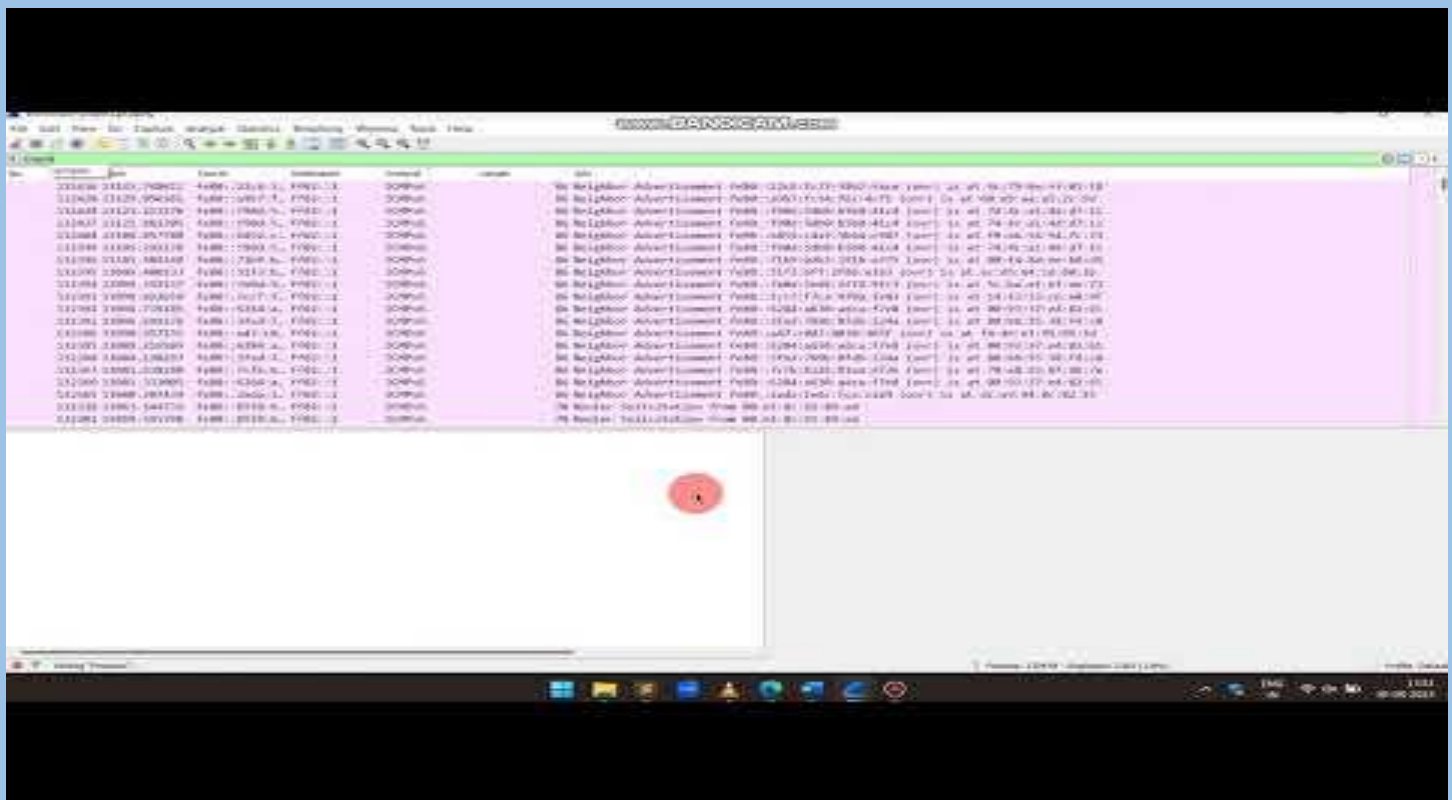
- SPAN works by copying the traffic from one or more source ports. The copy is then sent out a SPAN destination port. The destination port will often be connected to a host running packet analyzing software, such as Wireshark. Because SPAN only makes a copy of traffic, the source traffic is never affected.
- So, Source Port [the device for which packets are to be captured] traffic is captured and sent to SPAN destination Port [On which Wireshark is running].



→ PoC's:

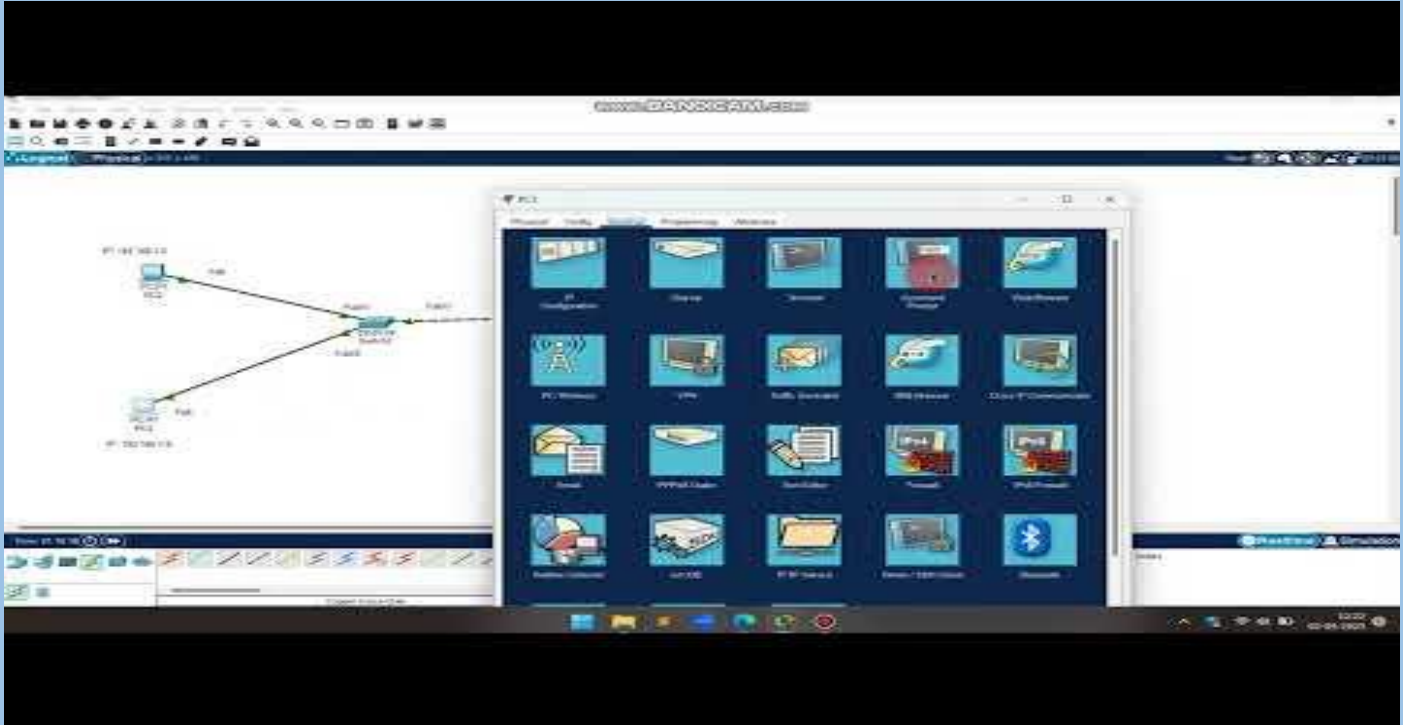
1	3 Way Handshake	Refer File “3-way-handshake-SPAN-SWITCH.pcapng” and Point-7 For 3-way handshake explanation.
2	TCP packet Analysis and exploring the features in the packet header window. [TCP header and IP header details for the selected packet.]	Refer Point-8 Explanation above. Also Refer the File “TCP-Header-Analysis.pcapng” to see the captured packets.

3. Other parts such as different protocols, different Filters, total number of captured packets for each protocol and exploring the Follow TCP stream feature in Wireshark. [Refer “Wireshark-Project-PoC.mp4” in Video-PoC Folder]



<https://youtu.be/LUOQi2nAcn0>

4. Port Mirroring simulation using Cisco-Packet Tracer [Refer “Packet-Sniffing-Simulation-via-port-mirroring (SPAN).mp4” in Video-PoC Folder]



https://youtu.be/1FgHt_Oi3Ms

→ Team Members:

1.	SHAIENDRA SINGH SACHAN	23030241186
2.	ADITI PANDEY	23030241148
3.	MOHIT RATHOUR	23030241174
4.	RIYA JAIN	23030241087
5.	HRUSHIKESH THATE	23030241047
6.	ASHITA GADE	23030241150