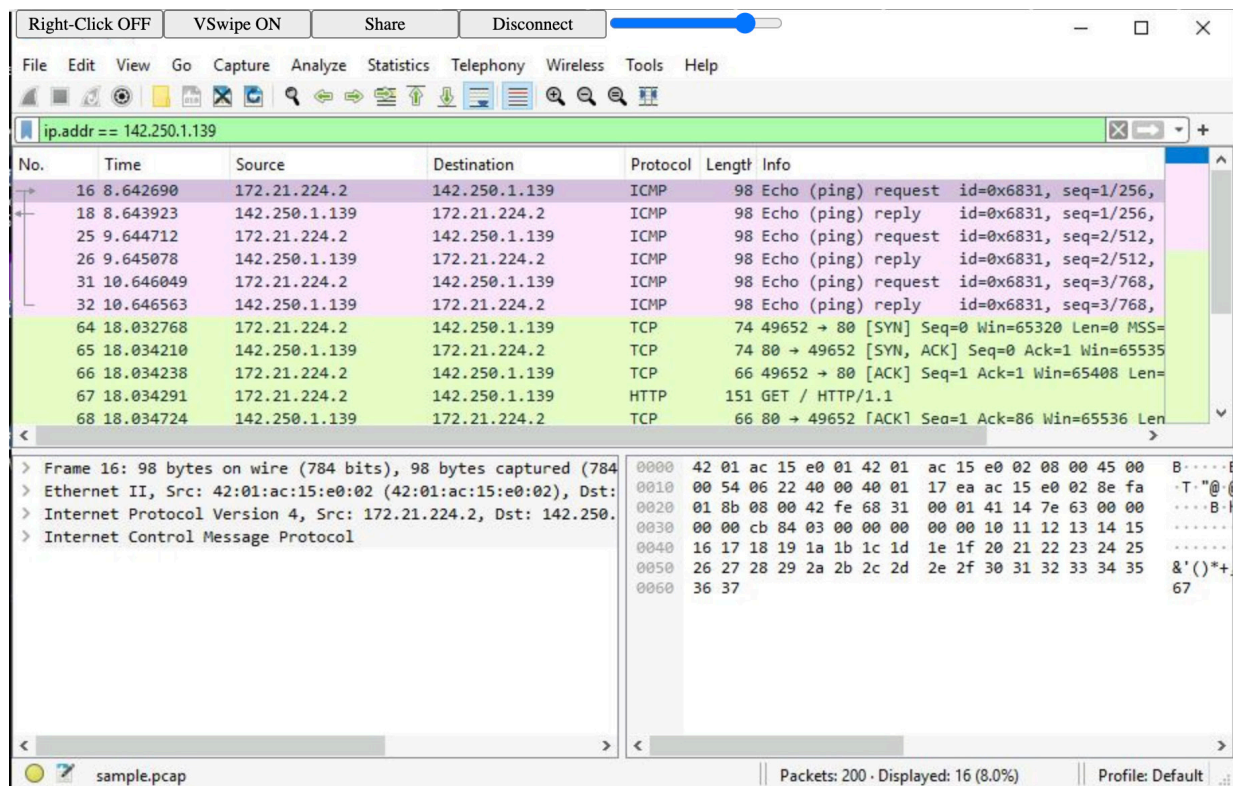


Project

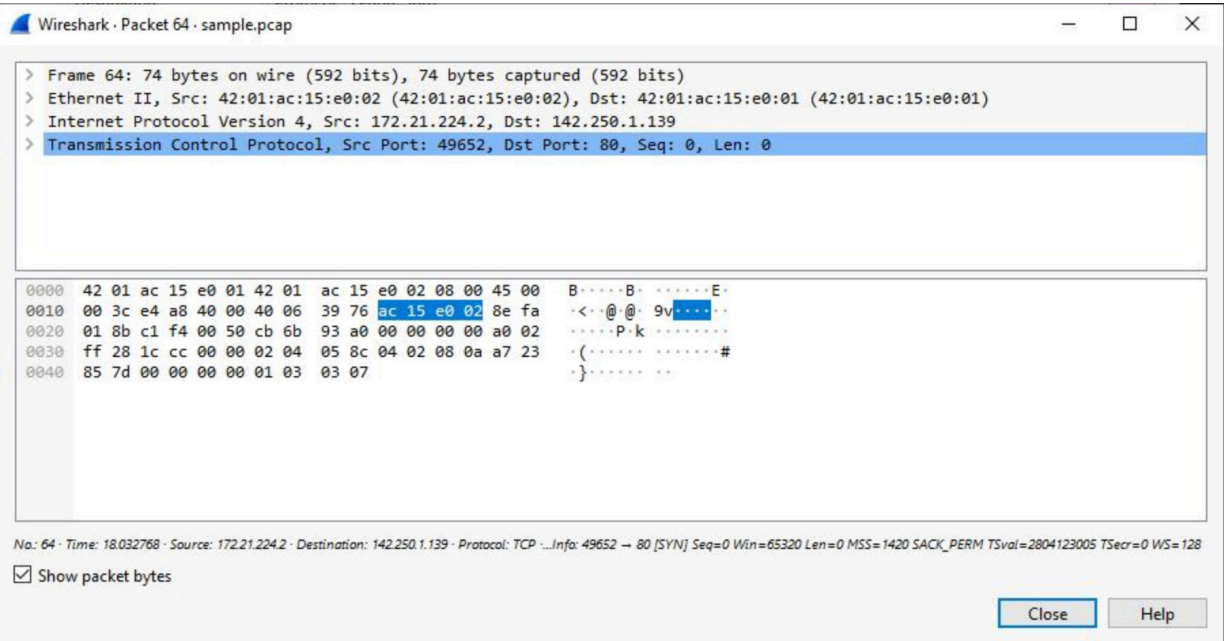
In this project, I will analyze a network packet capture file to determine the source and destination IP addresses involved in a web browsing session. I will investigate the protocols used during the connection to the website, such as TCP, HTTP, or DNS. Additionally, I will inspect specific data packets to identify the type of information being transmitted between systems, including headers, payloads, and metadata

To accomplish this, I will employ tools like Wireshark or tcpdump to capture and filter network traffic effectively. These tools will enable me to dissect packets by applying filters, such as IP address or protocol-specific filters, for focused analysis. Through this project, I aim to gain insights into how data flows across the network during a browsing session, highlighting communication patterns and potential anomalies. Ultimately, this experience will enhance my skills in network traffic analysis and improve my ability to monitor and interpret traffic for operational and security purposes.

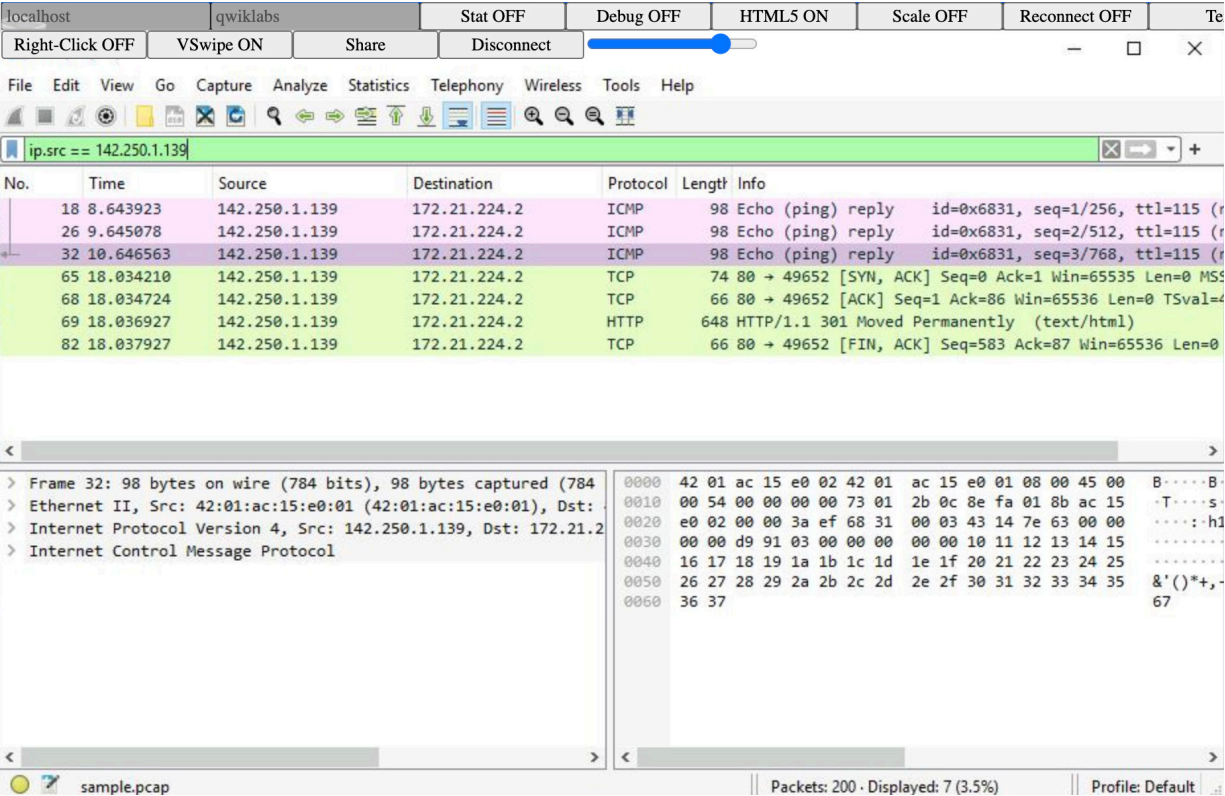
I opened a packet in Wireshark for detailed exploration and applied a display filter to inspect the network layers and protocols contained within the packet. Specifically, I entered the command `ip.addr == 142.250.1.139` in the "Apply a display filter..." text box located just above the packet list. This filter allowed me to isolate and view all traffic associated with that specific IP address, enabling a focused analysis of the relevant packets.



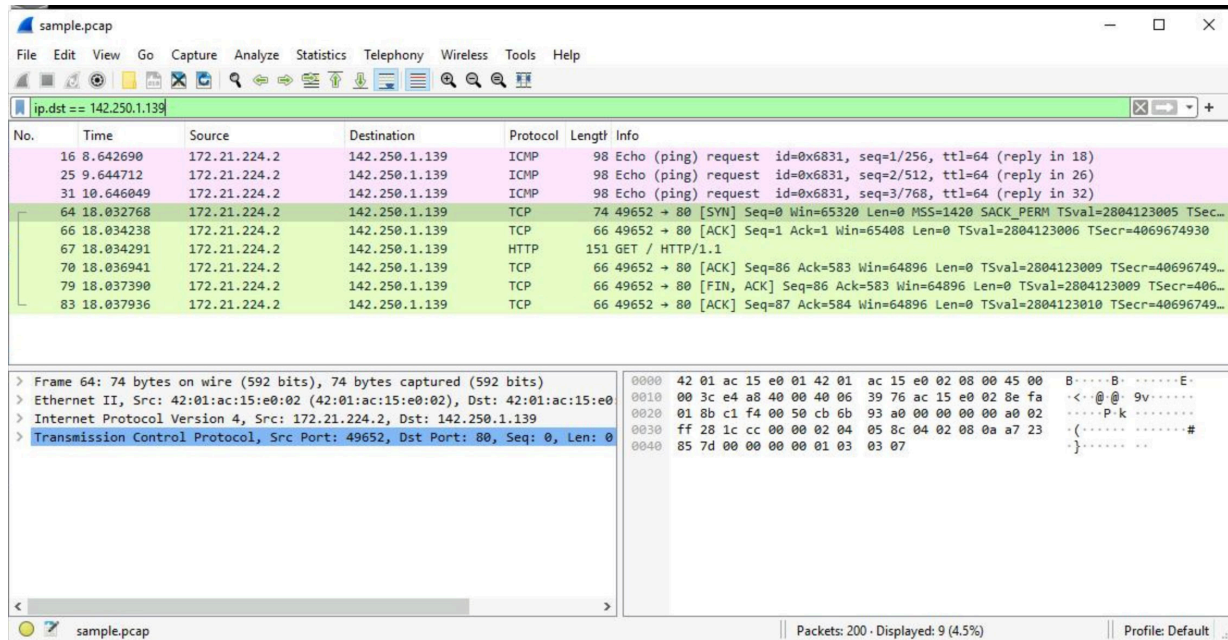
I was then tasked with identifying the destination port of the first TCP packet. To do this, I double-clicked on the Transmission Control Protocol subtree, which revealed that the destination port is 80.



In this task, I used the command `ip.src == 142.250.1.139` to filter and analyze specific network packets based on their source or destination. I explored how to select packets using either their physical Ethernet Media Access Control (MAC) address or their Internet Protocol (IP) address.



I entered the following filter to select traffic for a specific destination IP address: `ip.dst == 142.250.1.139`.



sample.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

`ip.dst == 142.250.1.139`

No.	Time	Source	Destination	Protocol	Length	Info
16	8.642690	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=1/256, ttl=64 (reply in 18)
25	9.644712	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=2/512, ttl=64 (reply in 26)
31	10.646049	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=3/768, ttl=64 (reply in 32)
64	18.032768	172.21.224.2	142.250.1.139	TCP	74	49652 → 80 [SYN] Seq=0 Win=65520 Len=0 MSS=1420 SACK_PERM TSval=2804123005 TSecr=...
66	18.034238	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [ACK] Seq=1 Ack=1 Win=65408 Len=0 TSval=2804123006 TSecr=4069674930
67	18.034291	172.21.224.2	142.250.1.139	HTTP	151	GET / HTTP/1.1
70	18.036941	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [ACK] Seq=86 Ack=583 Win=64896 Len=0 TSval=2804123009 TSecr=40696749...
79	18.037390	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [FIN, ACK] Seq=86 Ack=583 Win=64896 Len=0 TSval=2804123009 TSecr=406...
83	18.037936	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [ACK] Seq=87 Ack=584 Win=64896 Len=0 TSval=2804123010 TSecr=40696749...

> Frame 64: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)

> Ethernet II, Src: 42:01:ac:15:e0:02 (42:01:ac:15:e0:02), Dst: 42:01:ac:15:e0:02

> Internet Protocol Version 4, Src: 172.21.224.2, Dst: 142.250.1.139

> Transmission Control Protocol, Src Port: 49652, Dst Port: 80, Seq: 0, Len: 0

0000 42 01 ac 15 e0 01 42 01 ac 15 e0 02 00 00 45 00 B.....B.....E..

0010 00 3c e4 a8 40 00 40 06 39 76 ac 15 e0 02 8e fa <...@...9v.....

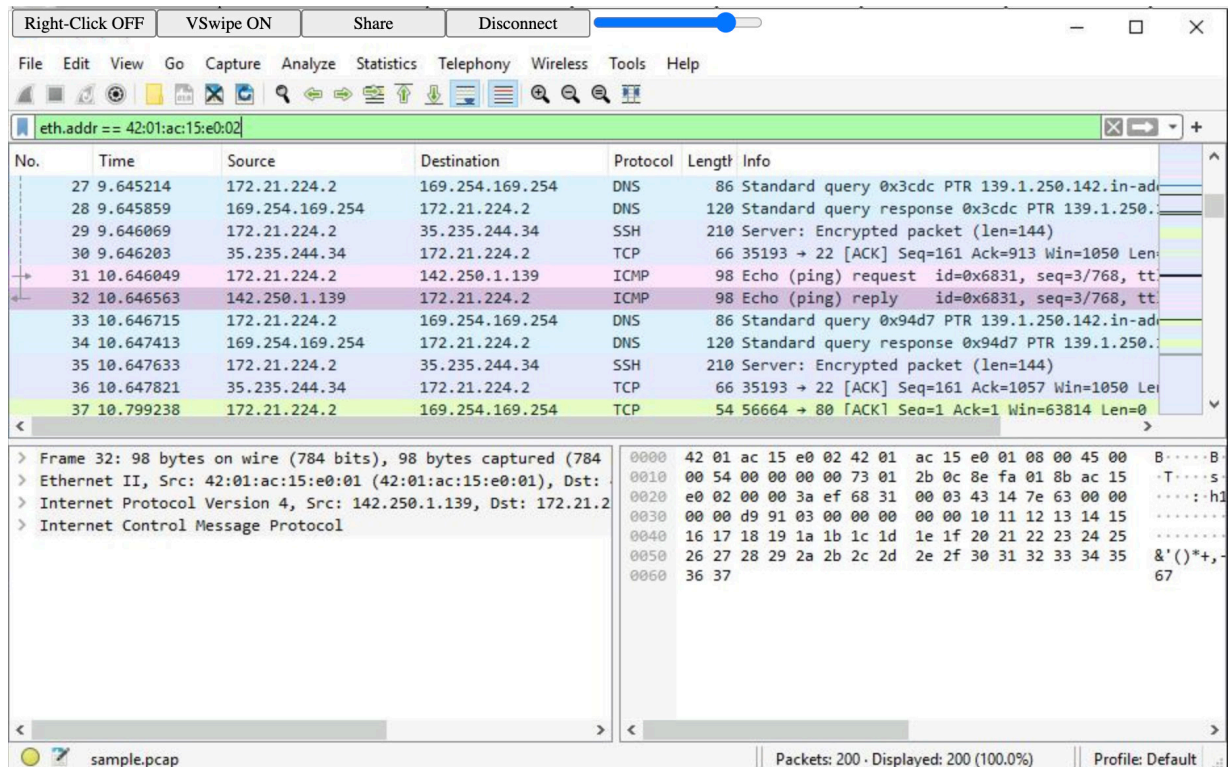
0020 01 8b c1 f4 00 50 cb 6b 93 a0 00 00 00 a0 02P-k.....

0030 ff 28 1c cc 00 02 04 05 8c 04 02 08 0a a7 23 {...}.....#.....

0040 85 7d 00 00 00 00 01 03 03 07}.....

sample.pcap Packets: 200 · Displayed: 9 (4.5%) Profile: Default

I used the command `eth.addr == 42:01:ac:15:e0:02` to select traffic to or from a specific Ethernet MAC address.



Right-Click OFF VSwipe ON Share Disconnect

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

`eth.addr == 42:01:ac:15:e0:02`

No.	Time	Source	Destination	Protocol	Length	Info
27	9.645214	172.21.224.2	169.254.169.254	DNS	86	Standard query 0x3cdc PTR 139.1.250.142.in-ad...
28	9.645859	169.254.169.254	172.21.224.2	DNS	120	Standard query response 0x3cdc PTR 139.1.250...
29	9.646069	172.21.224.2	35.235.244.34	SSH	210	Server: Encrypted packet (len=144)
30	9.646203	35.235.244.34	172.21.224.2	TCP	66	35193 → 22 [ACK] Seq=161 Ack=913 Win=1050 Len=...
31	10.646049	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=3/768, tt...
32	10.646563	142.250.1.139	172.21.224.2	ICMP	98	Echo (ping) reply id=0x6831, seq=3/768, tt...
33	10.646715	172.21.224.2	169.254.169.254	DNS	86	Standard query 0x94d7 PTR 139.1.250.142.in-ad...
34	10.647413	169.254.169.254	172.21.224.2	DNS	120	Standard query response 0x94d7 PTR 139.1.250...
35	10.647633	172.21.224.2	35.235.244.34	SSH	210	Server: Encrypted packet (len=144)
36	10.647821	35.235.244.34	172.21.224.2	TCP	66	35193 → 22 [ACK] Seq=161 Ack=1057 Win=1050 Len=...
37	10.799238	172.21.224.2	169.254.169.254	TCP	54	56664 → 80 [ACK] Seq=1 Ack=1 Win=63814 Len=0

> Frame 32: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)

> Ethernet II, Src: 42:01:ac:15:e0:01 (42:01:ac:15:e0:01), Dst: 42:01:ac:15:e0:02

> Internet Protocol Version 4, Src: 142.250.1.139, Dst: 172.21.224.2

> Internet Control Message Protocol

0000 42 01 ac 15 e0 02 42 01 ac 15 e0 01 08 00 45 00 B.....B.....T.....

0010 00 54 00 00 00 00 73 01 2b 0c 8e fa 01 8b ac 15T.....s.....

0020 e0 02 00 00 3a ef 68 31 00 03 43 14 7e 63 00 00:..h1.....

0030 00 00 d9 91 03 00 00 00 00 00 10 11 12 13 14 15:.....

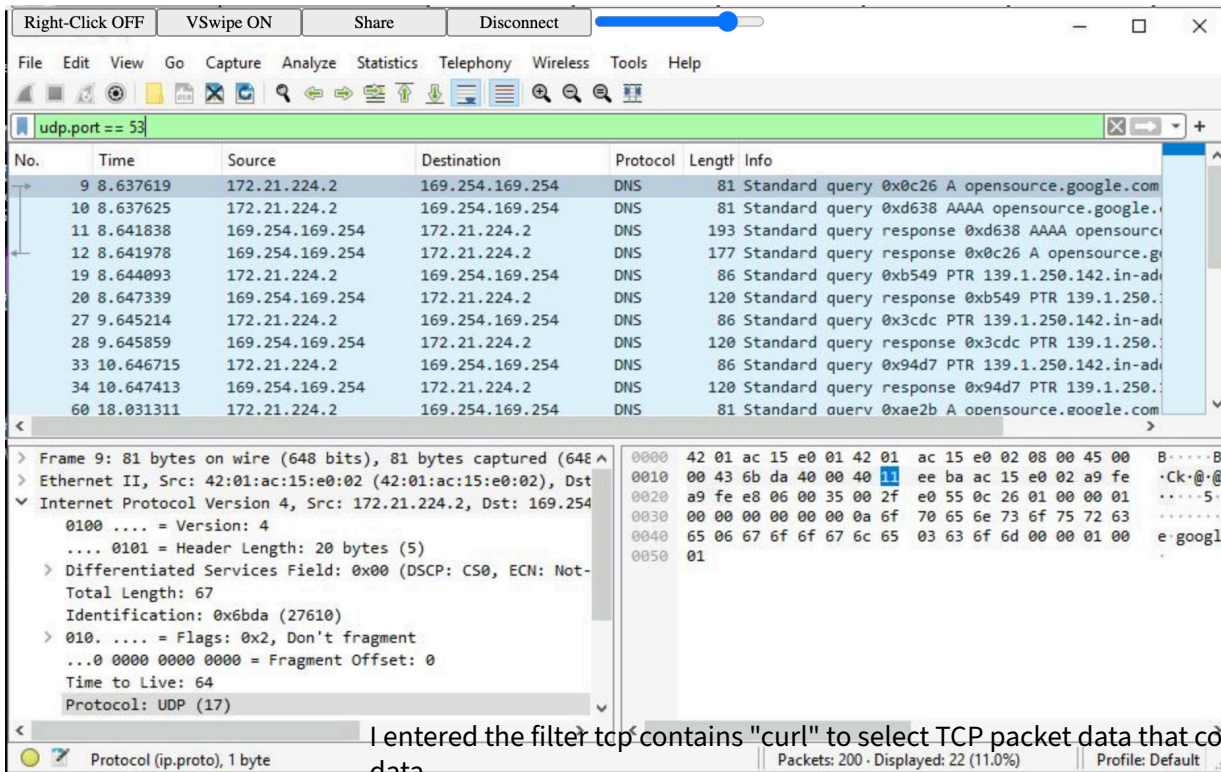
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25:.....

0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35&'()*+,-.....

0060 36 3767.....

sample.pcap Packets: 200 · Displayed: 200 (100.0%) Profile: Default

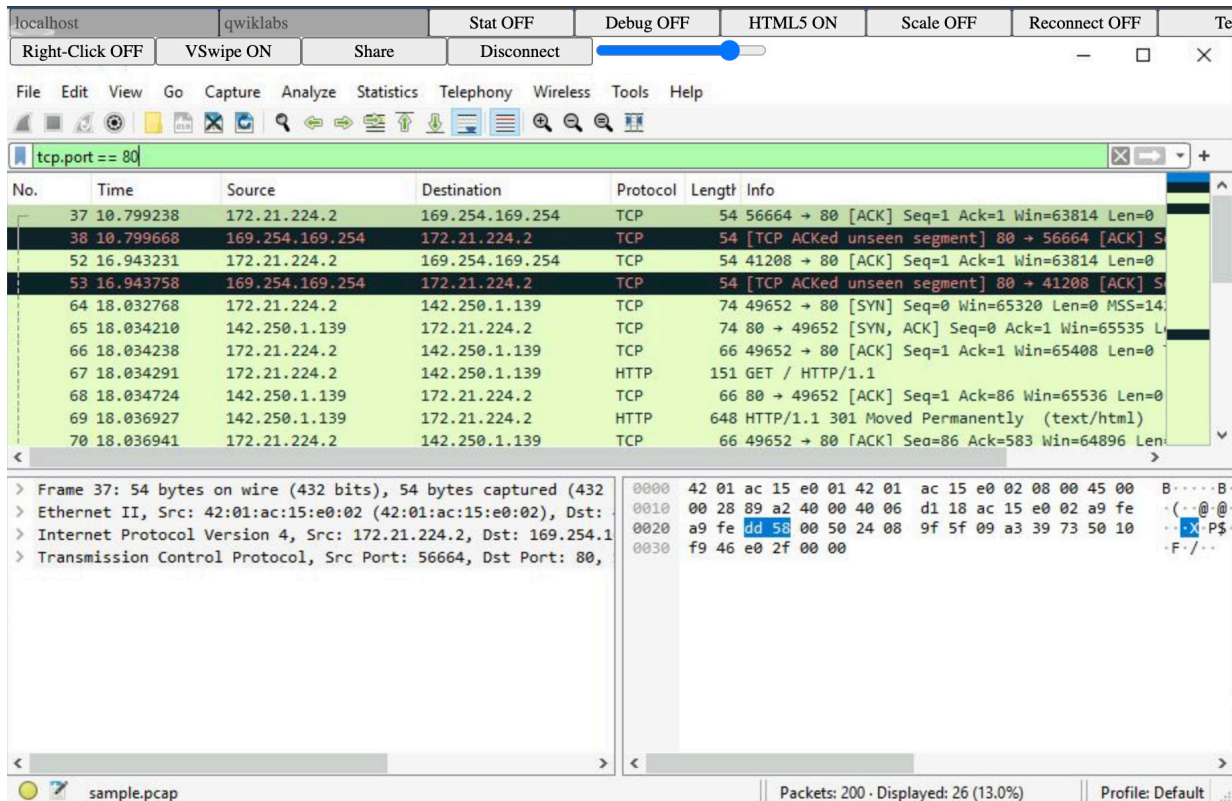
I entered the filter `udp.port == 53` to select UDP port 53 traffic, which is used for DNS queries and responses:



The screenshot shows the Wireshark interface with the filter `udp.port == 53` applied. The packet list displays several DNS queries and responses between 172.21.224.2 and 169.254.169.254. The packet details pane for Frame 9 shows an Ethernet II header, an Internet Protocol Version 4 header, and a UDP header. The packet bytes pane shows the raw data of the packet.

I entered the filter `tcp contains "curl"` to select TCP packet data that contains specific text data.

I entered the filter `tcp.port == 80` to select TCP port 80 traffic, which is the default port associated with web traffic.



The screenshot shows the Wireshark interface with the filter `tcp.port == 80` applied. The packet list displays several TCP and HTTP packets between 172.21.224.2 and 142.250.1.139. The packet details pane for Frame 37 shows an Ethernet II header, an Internet Protocol Version 4 header, and a Transmission Control Protocol header. The packet bytes pane shows the raw data of the packet.

I entered the filter `tcp contains "curl"` to select TCP packet data that contains specific text data.

The image shows the Wireshark network packet analyzer interface. At the top, there are buttons for 'Right-Click OFF', 'VSwipe ON', 'Share', and 'Disconnect', along with a volume slider. Below these is a menu bar with 'File', 'Edit', 'View', 'Go', 'Capture', 'Analyze', 'Statistics', 'Telephony', 'Wireless', 'Tools', and 'Help'. A toolbar with various icons is positioned below the menu. The main display area is divided into three panes. The top pane shows a list of packets with columns for 'No.', 'Time', 'Source', 'Destination', 'Protocol', 'Length', and 'Info'. Two packets are selected and highlighted in green: packet 67 (HTTP GET) and packet 148 (HTTP GET). The middle pane shows the packet details for the selected packet, including 'Frame 67: 151 bytes on wire (1208 bits), 151 bytes captured (1208 bits)', 'Ethernet II', 'Internet Protocol Version 4', and 'Hypertext Transfer Protocol'. The bottom pane shows the raw packet data in hexadecimal and ASCII. The status bar at the bottom indicates 'sample.pcap', 'Packets: 200 · Displayed: 2 (1.0%)', and 'Profile: Default'.

No.	Time	Source	Destination	Protocol	Length	Info
67	18.034291	172.21.224.2	142.250.1.139	HTTP	151	GET / HTTP/1.1
148	42.369093	172.21.224.2	142.250.1.102	HTTP	151	GET / HTTP/1.1