

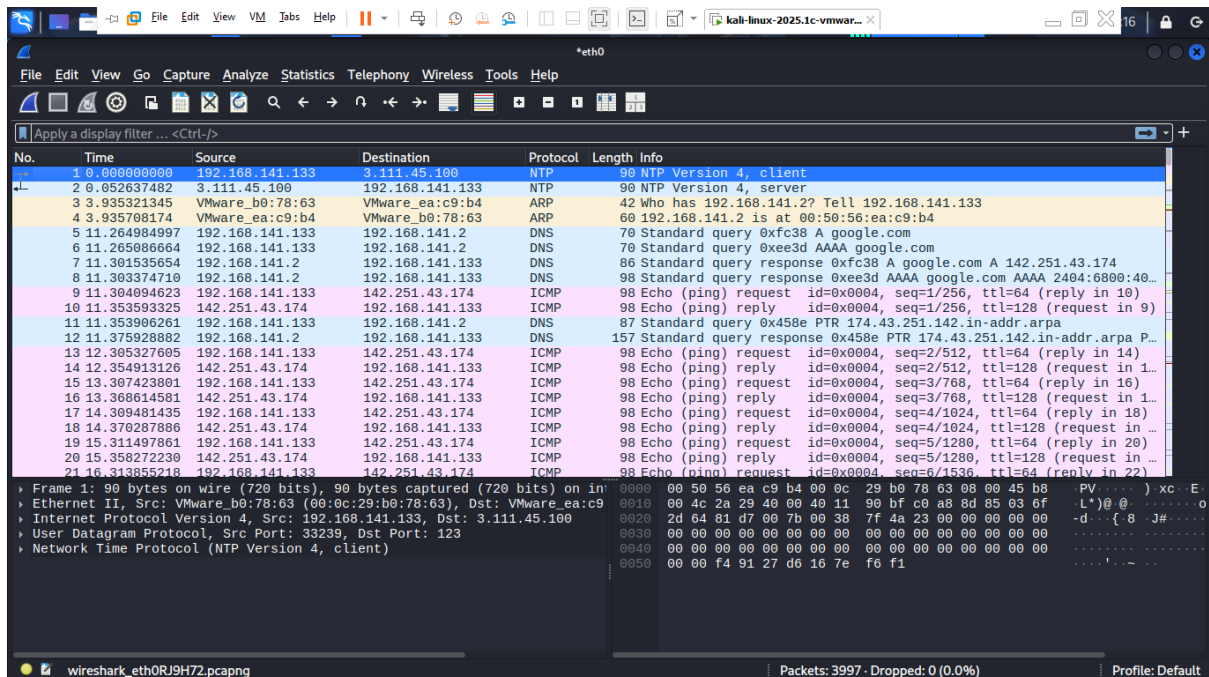
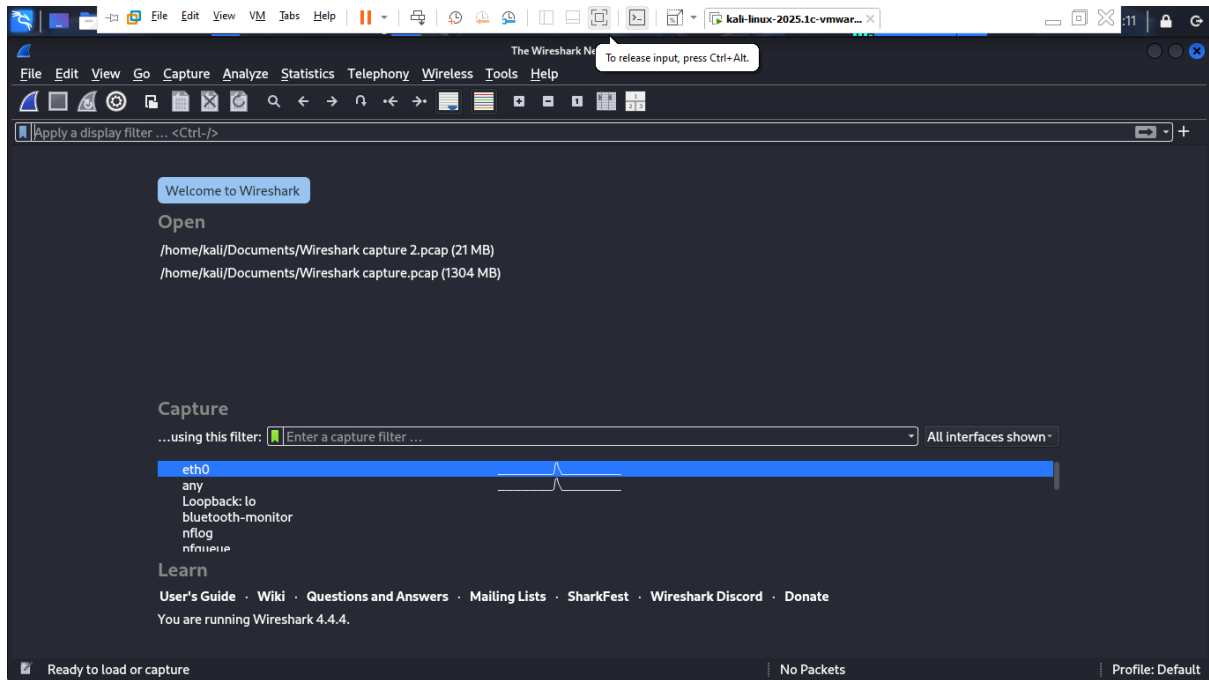
1. Install Wireshark

Command used : `sudo apt update`

`sudo apt install wireshark`

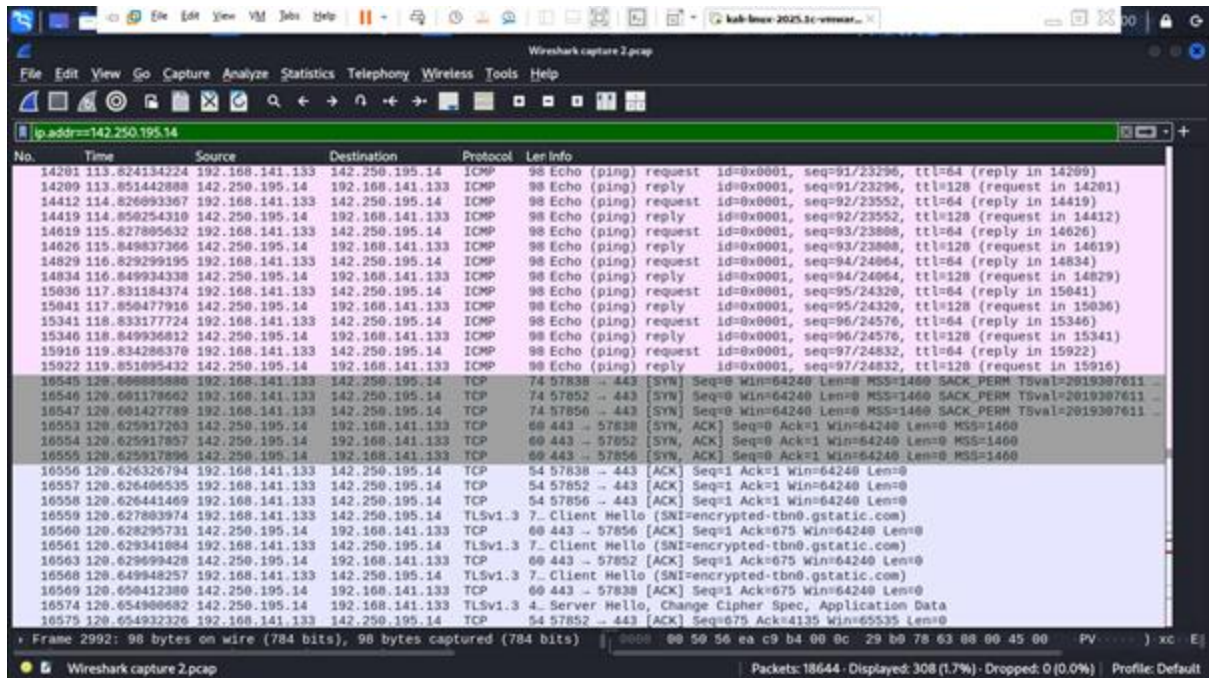
Here, in our case, wireshark is pre-installed in Kali Linux

2. Start capturing on active network interface



3. Browse a website or ping a website

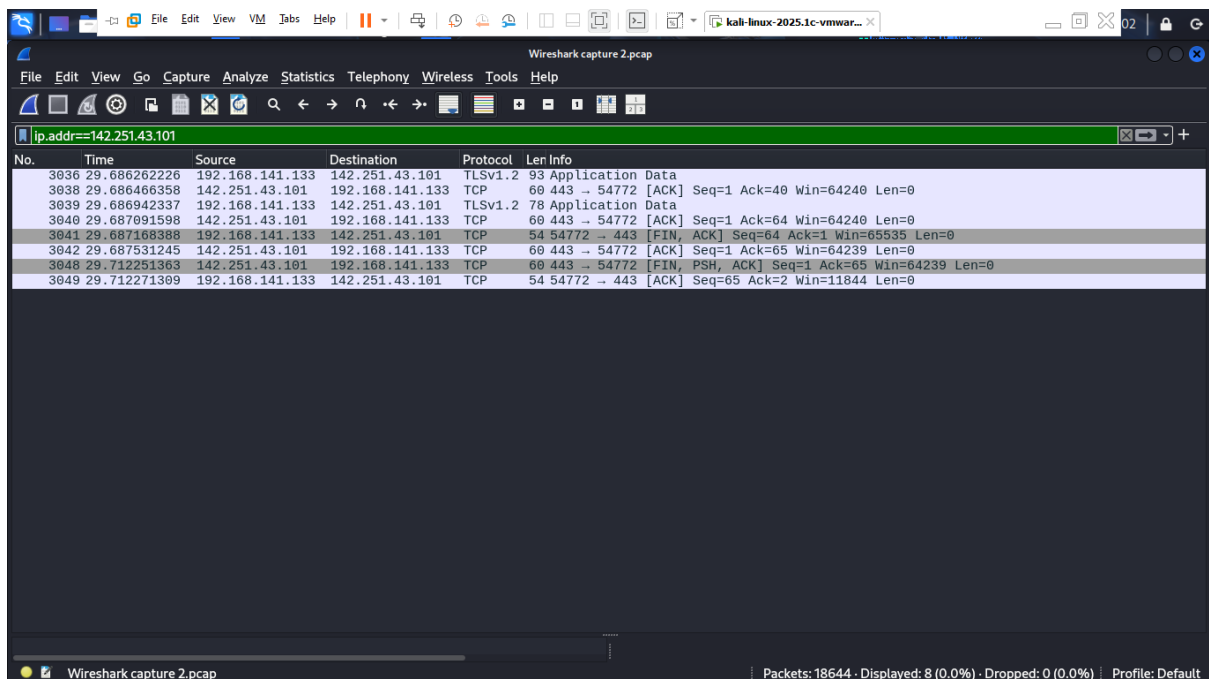
3.1 Ping request for google.com



The image shows a Wireshark capture of ICMP Echo (ping) requests and replies. The filter is set to 'ip.addr==142.250.195.14'. The packet list shows 19 packets, with the first 18 being Echo requests and the last one being an Echo reply. The packet details pane shows the structure of an ICMP Echo request, including the type, code, identifier, and sequence number. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Len	Info
14201	113.824134224	192.168.141.133	142.250.195.14	ICMP	98	Echo (ping) request id=0x0001, seq=91/23296, ttl=64 (reply in 14209)
14209	113.851442888	142.250.195.14	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0001, seq=91/23296, ttl=128 (request in 14201)
14412	114.826893367	192.168.141.133	142.250.195.14	ICMP	98	Echo (ping) request id=0x0001, seq=92/23552, ttl=64 (reply in 14419)
14419	114.856254319	142.250.195.14	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0001, seq=92/23552, ttl=128 (request in 14412)
14619	115.827895632	192.168.141.133	142.250.195.14	ICMP	98	Echo (ping) request id=0x0001, seq=93/23808, ttl=64 (reply in 14626)
14626	115.849837366	142.250.195.14	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0001, seq=93/23808, ttl=128 (request in 14619)
14829	116.829299195	192.168.141.133	142.250.195.14	ICMP	98	Echo (ping) request id=0x0001, seq=94/24064, ttl=64 (reply in 14834)
14834	116.849934338	142.250.195.14	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0001, seq=94/24064, ttl=128 (request in 14829)
15036	117.831184374	192.168.141.133	142.250.195.14	ICMP	98	Echo (ping) request id=0x0001, seq=95/24320, ttl=64 (reply in 15041)
15041	117.850477916	142.250.195.14	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0001, seq=95/24320, ttl=128 (request in 15036)
15341	118.833177724	192.168.141.133	142.250.195.14	ICMP	98	Echo (ping) request id=0x0001, seq=96/24576, ttl=64 (reply in 15346)
15346	118.849936812	142.250.195.14	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0001, seq=96/24576, ttl=128 (request in 15341)
15916	119.834286370	192.168.141.133	142.250.195.14	ICMP	98	Echo (ping) request id=0x0001, seq=97/24832, ttl=64 (reply in 15922)
15922	119.851095432	142.250.195.14	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0001, seq=97/24832, ttl=128 (request in 15916)
16545	120.688885886	192.168.141.133	142.250.195.14	TCP	74	57838 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2019307611
16546	120.681178662	192.168.141.133	142.250.195.14	TCP	74	57852 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2019307611
16547	120.681427789	192.168.141.133	142.250.195.14	TCP	74	57856 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2019307611
16553	120.625917263	142.250.195.14	192.168.141.133	TCP	60	443 → 57838 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
16554	120.625917857	142.250.195.14	192.168.141.133	TCP	60	443 → 57852 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
16555	120.625917896	142.250.195.14	192.168.141.133	TCP	60	443 → 57856 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
16556	120.626326794	192.168.141.133	142.250.195.14	TCP	54	57838 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
16557	120.626406535	192.168.141.133	142.250.195.14	TCP	54	57852 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
16558	120.626441469	192.168.141.133	142.250.195.14	TCP	54	57856 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
16559	120.627803974	192.168.141.133	142.250.195.14	TLSv1.3	7	Client Hello (SNI=encrypted-tbn0.gstatic.com)
16560	120.628295731	142.250.195.14	192.168.141.133	TCP	60	443 → 57856 [ACK] Seq=1 Ack=675 Win=64240 Len=0
16561	120.629341084	192.168.141.133	142.250.195.14	TLSv1.3	7	Client Hello (SNI=encrypted-tbn0.gstatic.com)
16563	120.629699428	142.250.195.14	192.168.141.133	TCP	60	443 → 57852 [ACK] Seq=1 Ack=675 Win=64240 Len=0
16568	120.649848257	192.168.141.133	142.250.195.14	TLSv1.3	7	Client Hello (SNI=encrypted-tbn0.gstatic.com)
16569	120.650412380	142.250.195.14	192.168.141.133	TCP	60	443 → 57838 [ACK] Seq=1 Ack=675 Win=64240 Len=0
16574	120.654900682	142.250.195.14	192.168.141.133	TLSv1.3	4	Server Hello, Change Cipher Spec, Application Data
16575	120.654932326	192.168.141.133	142.250.195.14	TCP	54	57852 → 443 [ACK] Seq=675 Ack=4135 Win=65535 Len=0

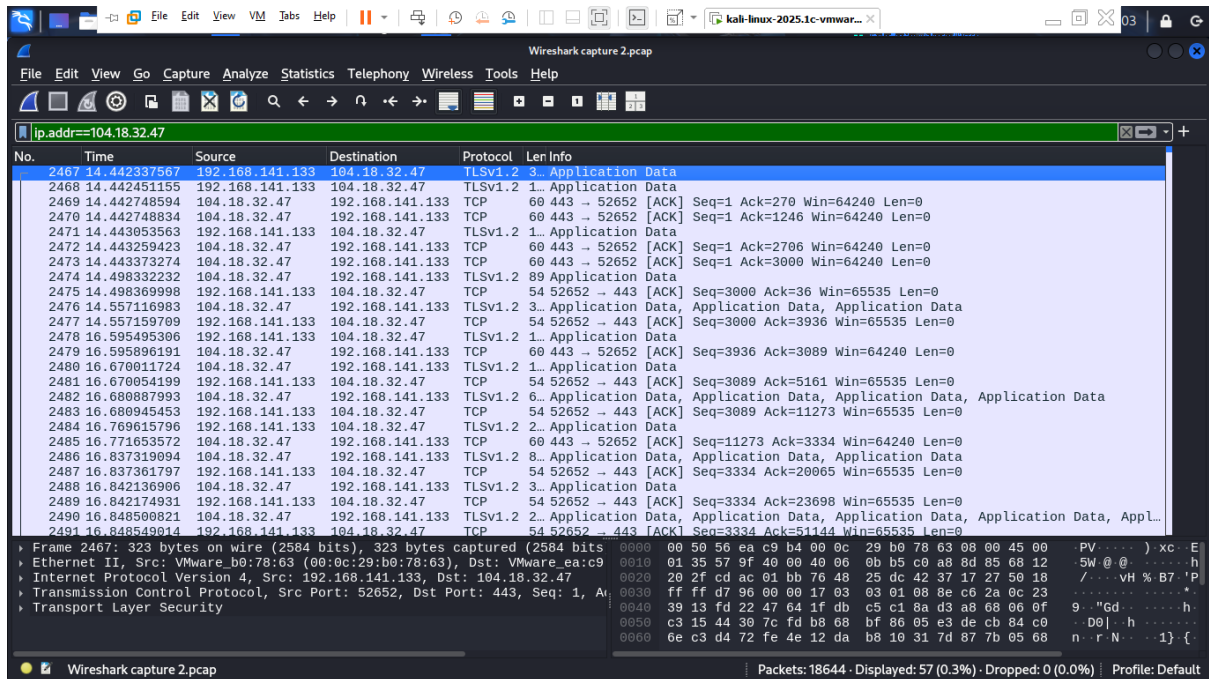
3.2 Ping request for gmail.com



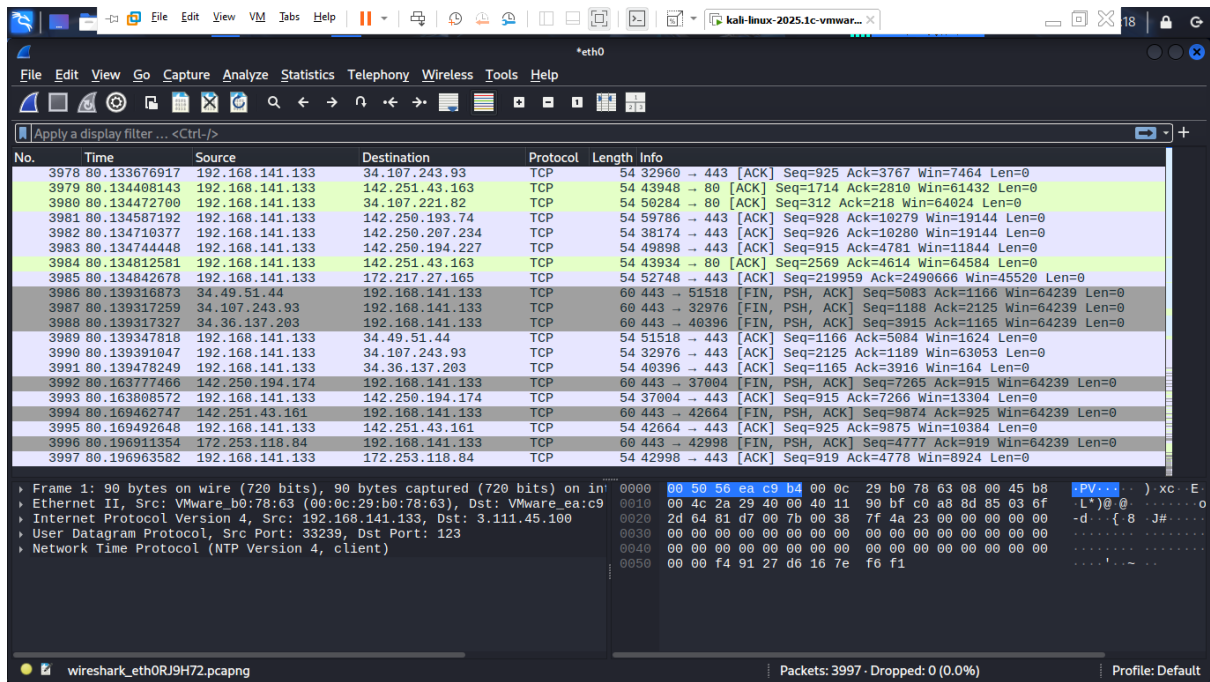
The image shows a Wireshark capture of a TLS handshake. The filter is set to 'ip.addr==142.251.43.101'. The packet list shows 8 packets, with the first 4 being TLS handshake messages and the last 4 being TCP ACKs. The packet details pane shows the structure of a TLS handshake, including the version, session ID, cipher suites, and compression methods. The packet bytes pane shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Len	Info
3036	29.686262226	192.168.141.133	142.251.43.101	TLSv1.2	93	Application Data
3038	29.686466358	142.251.43.101	192.168.141.133	TCP	60	443 → 54772 [ACK] Seq=1 Ack=40 Win=64240 Len=0
3039	29.686942337	192.168.141.133	142.251.43.101	TLSv1.2	78	Application Data
3040	29.687091598	142.251.43.101	192.168.141.133	TCP	60	443 → 54772 [ACK] Seq=1 Ack=64 Win=64240 Len=0
3041	29.687168388	192.168.141.133	142.251.43.101	TCP	54	54772 → 443 [FIN, ACK] Seq=64 Ack=1 Win=65535 Len=0
3042	29.687531245	142.251.43.101	192.168.141.133	TCP	60	443 → 54772 [ACK] Seq=1 Ack=65 Win=64239 Len=0
3048	29.712251363	142.251.43.101	192.168.141.133	TCP	60	443 → 54772 [FIN, PSH, ACK] Seq=1 Ack=65 Win=64239 Len=0
3049	29.712271309	192.168.141.133	142.251.43.101	TCP	54	54772 → 443 [ACK] Seq=65 Ack=2 Win=11844 Len=0

3.3 Ping request for chatgpt.com

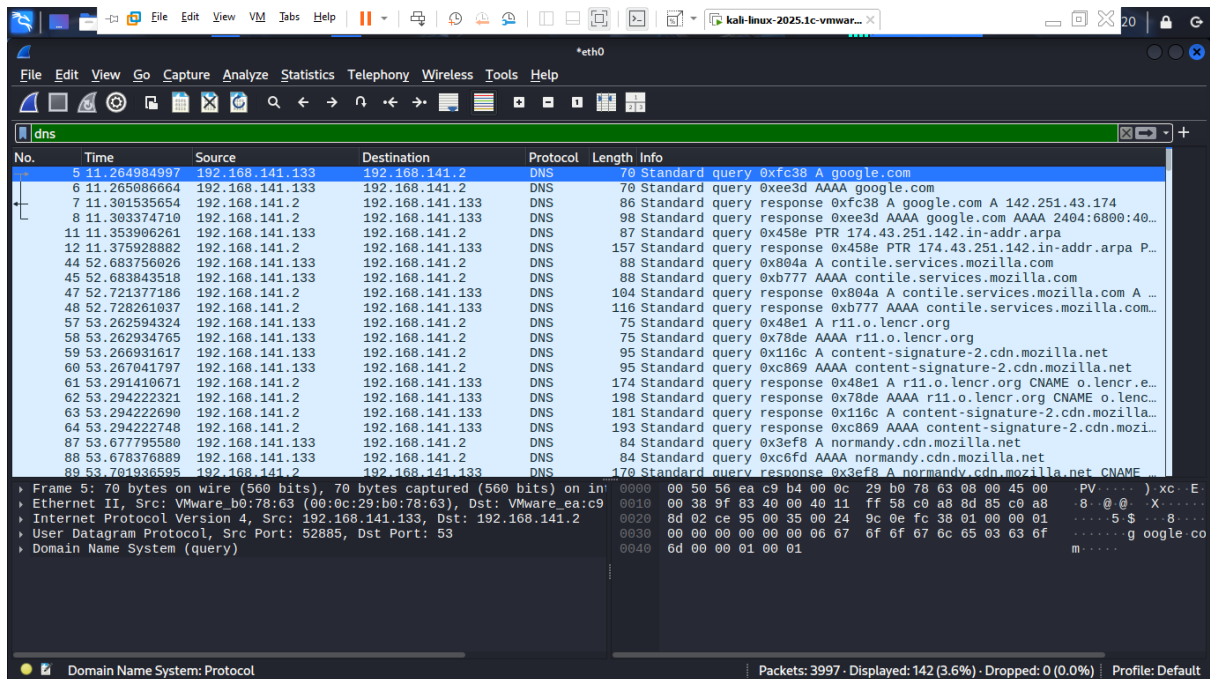


4. Stop capturing after sometime



5. Filter by different protocol type

5.1 DNS



5.2 ICMP

Wireshark capture of ICMP traffic on interface eth0. The packet list shows 28 ICMP Echo (ping) requests and replies between 192.168.141.133 and 142.251.43.174. The packet details pane shows the structure of an ICMP Echo (ping) reply, including the Ethernet II header, Internet Protocol Version 4 header, and Internet Control Message Protocol header. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
9	11.364894623	192.168.141.133	142.251.43.174	ICMP	98	Echo (ping) request id=0x0004, seq=1/256, ttl=64 (reply in 10)
10	11.353593325	142.251.43.174	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0004, seq=1/256, ttl=128 (request in 9)
13	12.305327605	192.168.141.133	142.251.43.174	ICMP	98	Echo (ping) request id=0x0004, seq=2/512, ttl=64 (reply in 14)
14	12.354913126	142.251.43.174	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0004, seq=2/512, ttl=128 (request in 13)
15	13.367423801	192.168.141.133	142.251.43.174	ICMP	98	Echo (ping) request id=0x0004, seq=3/768, ttl=64 (reply in 16)
16	13.368614581	142.251.43.174	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0004, seq=3/768, ttl=128 (request in 15)
17	14.369481435	192.168.141.133	142.251.43.174	ICMP	98	Echo (ping) request id=0x0004, seq=4/1024, ttl=64 (reply in 18)
18	14.370287886	142.251.43.174	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0004, seq=4/1024, ttl=128 (request in 17)
19	15.311497861	192.168.141.133	142.251.43.174	ICMP	98	Echo (ping) request id=0x0004, seq=5/1280, ttl=64 (reply in 20)
20	15.358272230	142.251.43.174	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0004, seq=5/1280, ttl=128 (request in 19)
21	16.313855218	192.168.141.133	142.251.43.174	ICMP	98	Echo (ping) request id=0x0004, seq=6/1536, ttl=64 (reply in 22)
22	16.368642224	142.251.43.174	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0004, seq=6/1536, ttl=128 (request in 21)
23	17.315652875	192.168.141.133	142.251.43.174	ICMP	98	Echo (ping) request id=0x0004, seq=7/1792, ttl=64 (reply in 24)
24	17.368547576	142.251.43.174	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0004, seq=7/1792, ttl=128 (request in 23)
25	18.316851346	192.168.141.133	142.251.43.174	ICMP	98	Echo (ping) request id=0x0004, seq=8/2048, ttl=64 (reply in 26)
26	18.375025990	142.251.43.174	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0004, seq=8/2048, ttl=128 (request in 25)
27	19.318619595	192.168.141.133	142.251.43.174	ICMP	98	Echo (ping) request id=0x0004, seq=9/2304, ttl=64 (reply in 28)
28	19.373532186	142.251.43.174	192.168.141.133	ICMP	98	Echo (ping) reply id=0x0004, seq=9/2304, ttl=128 (request in 27)

Frame 28: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface eth0
Ethernet II, Src: VMware ea:c9:b4 (00:50:56:ea:c9:b4), Dst: VMware b0:78:63 (00:50:56:b0:78:63)
Internet Protocol Version 4, Src: 142.251.43.174, Dst: 192.168.141.133
Internet Control Message Protocol

Internet Control Message Protocol: Protocol

Packets: 3997 · Displayed: 18 (0.5%) · Dropped: 0 (0.0%) · Profile: Default

5.3 TCP

Wireshark capture of TCP traffic on interface eth0. The packet list shows a sequence of TCP packets, including SYN, ACK, and RST. The packet details pane shows the structure of a TCP segment, including the Ethernet II header, Internet Protocol Version 4 header, and Transmission Control Protocol header. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
49	52.729036882	192.168.141.133	34.36.137.203	TCP	74	40396 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=...
50	52.748777543	34.36.137.203	192.168.141.133	TCP	60	443 → 40396 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
51	52.748831959	192.168.141.133	34.36.137.203	TCP	54	40396 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
52	52.844347075	192.168.141.133	34.36.137.203	TLSv1.3	730	Client Hello (SNI=contile.services.mozilla.com)
53	52.844621505	34.36.137.203	192.168.141.133	TCP	60	443 → 40396 [ACK] Seq=1 Ack=677 Win=64240 Len=0
54	52.952287695	34.36.137.203	192.168.141.133	TLSv1.3	3175	Server Hello, Change Cipher Spec, Application Data
55	52.952352370	192.168.141.133	34.36.137.203	TCP	54	40396 → 443 [ACK] Seq=677 Ack=3122 Win=64240 Len=0
65	53.296506336	192.168.141.133	49.44.175.24	TCP	74	44230 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=...
66	53.296789392	192.168.141.133	34.160.144.191	TCP	74	52412 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=...
67	53.320049092	49.44.175.24	192.168.141.133	TCP	60	80 → 44230 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
68	53.320097884	192.168.141.133	49.44.175.24	TCP	54	44230 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
69	53.320644481	192.168.141.133	49.44.175.24	OCSP	485	Request
70	53.321096061	49.44.175.24	192.168.141.133	TCP	60	80 → 44230 [ACK] Seq=1 Ack=432 Win=64240 Len=0
71	53.326478123	34.160.144.191	192.168.141.133	TCP	60	443 → 52412 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
72	53.326645804	192.168.141.133	34.160.144.191	TCP	54	52412 → 443 [RST] Seq=1 Win=0 Len=0
73	53.350174768	49.44.175.24	192.168.141.133	OCSP	943	Response
74	53.350242734	192.168.141.133	49.44.175.24	TCP	54	44230 → 80 [ACK] Seq=432 Ack=890 Win=63351 Len=0
75	53.551709172	192.168.141.133	34.36.137.203	TLSv1.3	118	Change Cipher Spec, Application Data
76	53.552163078	34.36.137.203	192.168.141.133	TCP	60	443 → 40396 [ACK] Seq=3122 Ack=741 Win=64240 Len=0
77	53.554884295	192.168.141.133	34.36.137.203	TLSv1.3	146	Application Data
78	53.555265094	34.36.137.203	192.168.141.133	TCP	60	443 → 40396 [ACK] Seq=3122 Ack=833 Win=64240 Len=0

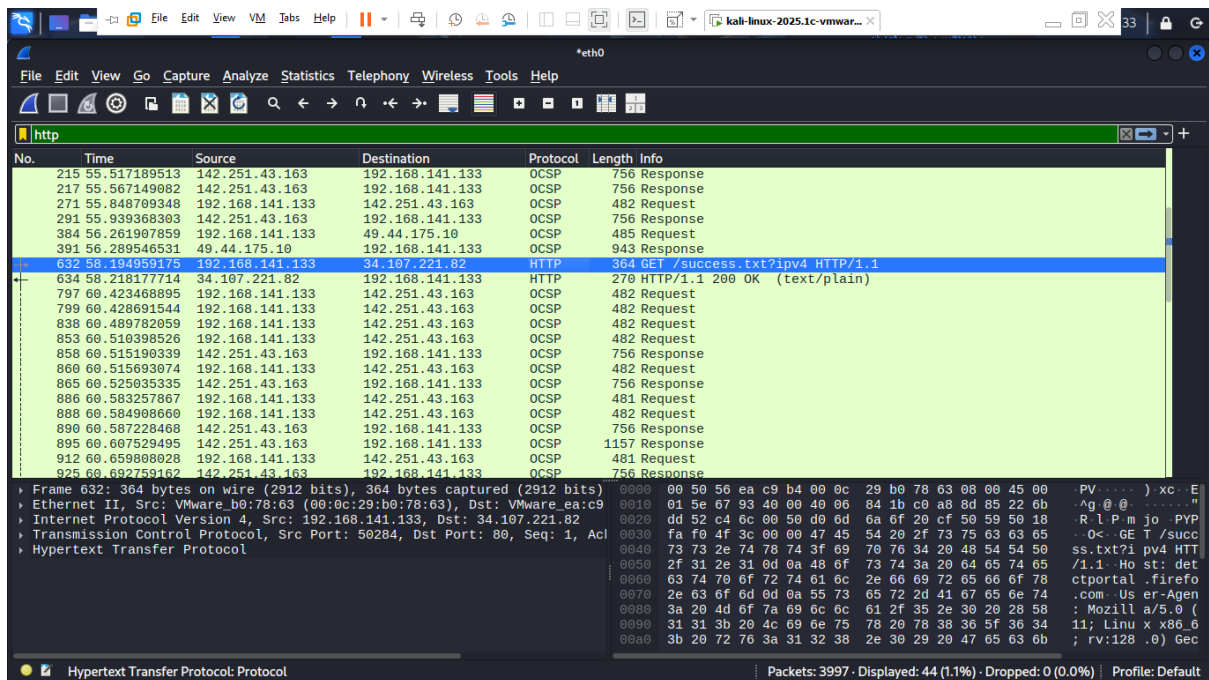
Frame 65: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0
Ethernet II, Src: VMware b0:78:63 (00:50:56:b0:78:63), Dst: VMware ea:c9:b4 (00:50:56:ea:c9:b4)
Internet Protocol Version 4, Src: 192.168.141.133, Dst: 49.44.175.24
Transmission Control Protocol, Src Port: 44230, Dst Port: 80, Seq: 0, Len: 0

Transmission Control Protocol: Protocol

Packets: 3997 · Displayed: 1796 (44.9%) · Dropped: 0 (0.0%) · Profile: Default

6. Identify at least 3 different protocols

6.1 HTTP

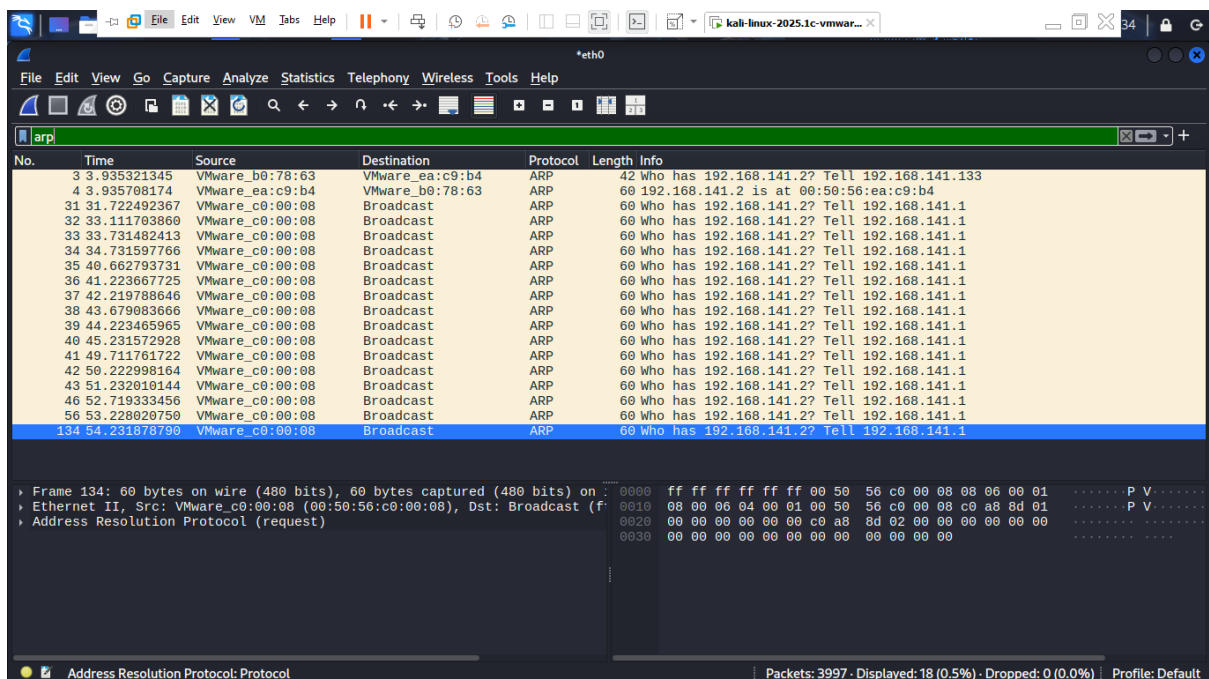


The screenshot shows a Wireshark capture of network traffic on the 'http' filter. The packet list on the left shows several packets, with packet 632 selected. The packet details pane on the right shows the structure of the selected packet, which is an HTTP GET request. The packet bytes pane at the bottom shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
215	55.517189513	142.251.43.163	192.168.141.133	OCSP	756	Response
217	55.567149982	142.251.43.163	192.168.141.133	OCSP	756	Response
271	55.848709348	192.168.141.133	142.251.43.163	OCSP	482	Request
291	55.939368303	142.251.43.163	192.168.141.133	OCSP	756	Response
384	56.261907859	192.168.141.133	49.44.175.10	OCSP	485	Request
391	56.289546531	49.44.175.10	192.168.141.133	OCSP	943	Response
632	58.194959175	192.168.141.133	34.107.221.82	HTTP	364	GET /success.txt?ipv4 HTTP/1.1
634	58.218177714	34.107.221.82	192.168.141.133	HTTP	270	HTTP/1.1 200 OK (text/plain)
797	60.423468895	192.168.141.133	142.251.43.163	OCSP	482	Request
799	60.428691544	192.168.141.133	142.251.43.163	OCSP	482	Request
838	60.489782059	192.168.141.133	142.251.43.163	OCSP	482	Request
853	60.510398526	192.168.141.133	142.251.43.163	OCSP	482	Request
858	60.515190339	142.251.43.163	192.168.141.133	OCSP	756	Response
860	60.515693074	192.168.141.133	142.251.43.163	OCSP	482	Request
865	60.525035335	142.251.43.163	192.168.141.133	OCSP	756	Response
886	60.583257867	192.168.141.133	142.251.43.163	OCSP	481	Request
888	60.584908660	192.168.141.133	142.251.43.163	OCSP	482	Request
890	60.587228468	142.251.43.163	192.168.141.133	OCSP	756	Response
895	60.607529495	142.251.43.163	192.168.141.133	OCSP	1157	Response
912	60.659080028	192.168.141.133	142.251.43.163	OCSP	481	Request
925	60.692759162	142.251.43.163	192.168.141.133	OCSP	756	Response

Frame 632: 364 bytes on wire (2912 bits), 364 bytes captured (2912 bits) on interface 0
Ethernet II, Src: VMware_b0:78:63 (00:0c:29:b0:78:63), Dst: VMware_ea:c9:b4:00:00:00
Internet Protocol Version 4, Src: 192.168.141.133, Dst: 34.107.221.82
Transmission Control Protocol, Src Port: 50284, Dst Port: 80, Seq: 1, Ack: 3410722182, Win: 0, Len: 0
Hypertext Transfer Protocol

6.2 ARP

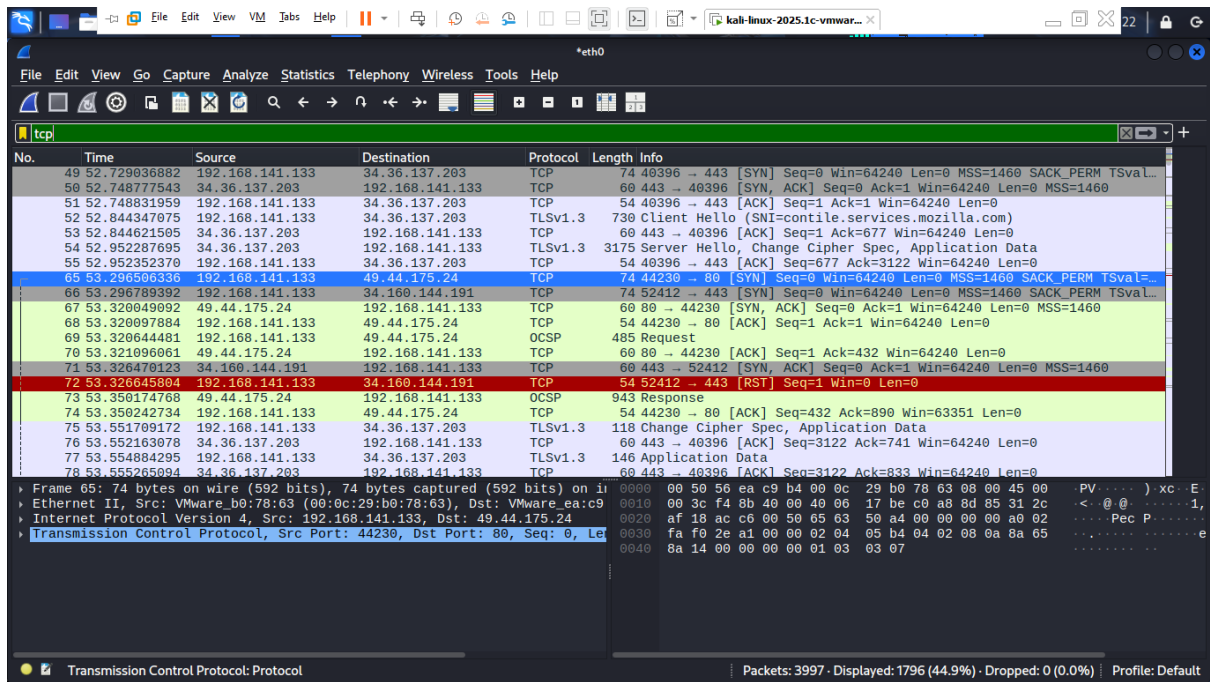


The screenshot shows a Wireshark capture of network traffic on the 'arp' filter. The packet list on the left shows several packets, with packet 134 selected. The packet details pane on the right shows the structure of the selected packet, which is an ARP request. The packet bytes pane at the bottom shows the raw data of the packet.

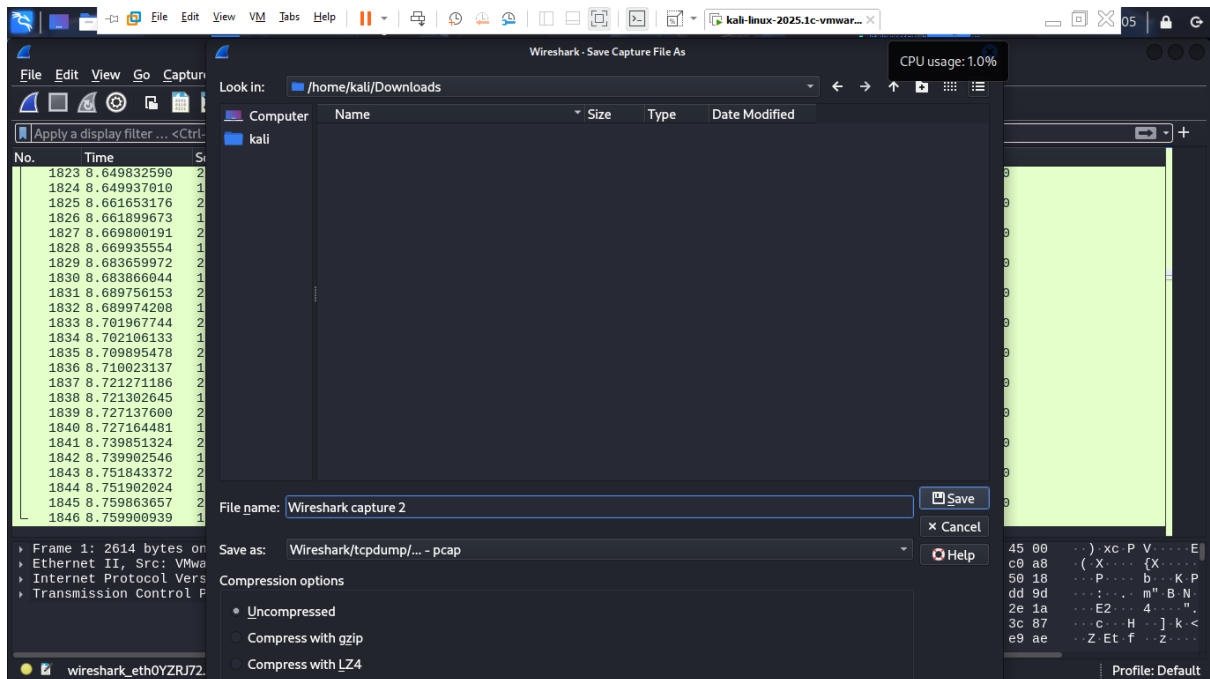
No.	Time	Source	Destination	Protocol	Length	Info
3	3.935321345	VMware_b0:78:63	VMware_ea:c9:b4:00:00:00	ARP	42	who has 192.168.141.2? Tell 192.168.141.133
4	3.935708174	VMware_ea:c9:b4:00:00:00	VMware_b0:78:63	ARP	60	192.168.141.2 is at 00:50:56:ea:c9:b4
31	31.722492367	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
32	31.111703860	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
33	33.731482413	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
34	34.731597766	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
35	40.662793731	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
36	41.223667725	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
37	42.219788646	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
38	43.679083666	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
39	44.223465965	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
40	45.231572928	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
41	49.711761722	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
42	50.222998164	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
43	51.232010144	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
46	52.719333456	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
56	53.228020750	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1
134	54.231878790	VMware_c0:00:00:00:00:00	Broadcast	ARP	60	who has 192.168.141.2? Tell 192.168.141.1

Frame 134: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: VMware_c0:00:00:00:00:00 (00:50:56:c0:00:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)

6.3 TCP



7. Exporting the packet capture as .pcap file



8. Summarize your findings and packet details.

- The Wireshark capture session successfully demonstrated the process of monitoring network activity on a Linux system.
- The capture included essential steps such as installing Wireshark, initiating a live capture on an active network interface, and generating traffic by pinging well-known websites like Google, Gmail, and ChatGPT. During the session, traffic was filtered based on protocol types such as DNS, ICMP, and TCP to observe how each behaves in a typical communication flow.
- Further inspection of the packet capture revealed additional network protocols like HTTPS and ARP, providing a broader view of the network interactions occurring during the session.
- The exercise culminated with the export of the capture into a .pcap file, offering a practical understanding of traffic analysis, protocol behaviour, and packet structure for network troubleshooting and educational purposes.