

Task 5 : Capture and Analyze Network Traffic Using Wireshark.

Step 1 :-

Install Wireshark on Linux Machine.

```
(root@kali)-[/home/kali]
# apt install wireshark -y
Upgrading:
  libwireshark-data libwireshark18 libwiretap15 libwsutil16 tshark wireshark wireshark-common

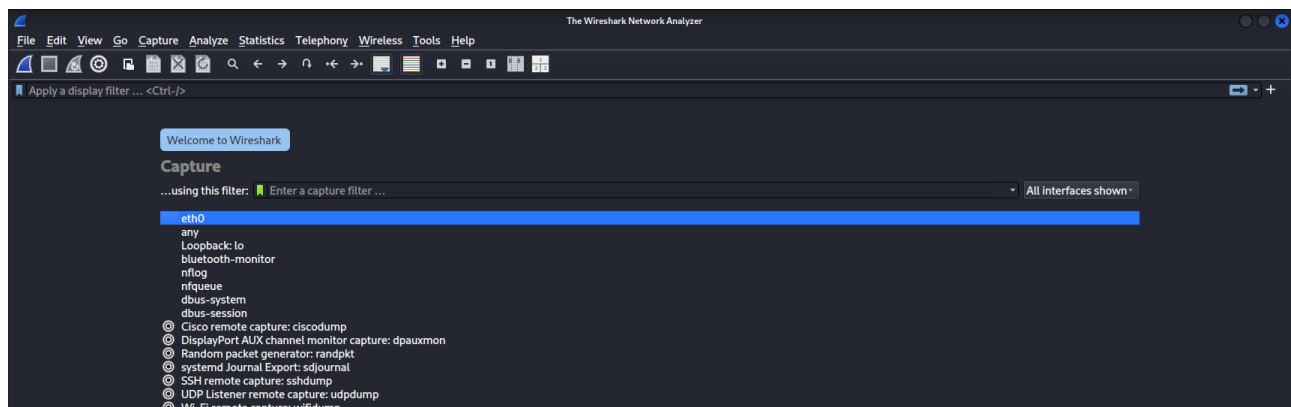
Summary:
  Upgrading: 7, Installing: 0, Removing: 0, Not Upgrading: 866
  Download size: 27.4 MB
  Space needed: 104 kB / 54.3 GB available

Get:1 http://kali.download/kali kali-rolling/main amd64 libwsutil16 amd64 4.4.9-1 [127 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 libwiretap15 amd64 4.4.9-1 [267 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 tshark amd64 4.4.9-1 [175 kB]
Get:5 http://kali.download/kali kali-rolling/main amd64 wireshark amd64 4.4.9-1 [4,626 kB]
Get:6 http://kali.download/kali kali-rolling/main amd64 wireshark-common amd64 4.4.9-1 [598 kB]
Get:3 http://mirrors.esto.network/kali kali-rolling/main amd64 libwireshark18 amd64 4.4.9-1 [21.2 MB]
Get:7 http://mirrors.esto.network/kali kali-rolling/main amd64 libwireshark-data all 4.4.9-1 [449 kB]
Fetched 27.4 MB in 19s (1,419 kB/s)
Preconfiguring packages ...
(Reading database ... 417523 files and directories currently installed.)
Preparing to unpack .../0-libwsutil16_4.4.9-1_amd64.deb ...
Unpacking libwsutil16:amd64 (4.4.9-1) over (4.4.7-1+b1) ...
Preparing to unpack .../1-libwiretap15_4.4.9-1_amd64.deb ...
Unpacking libwiretap15:amd64 (4.4.9-1) over (4.4.7-1+b1) ...
Preparing to unpack .../2-libwireshark18_4.4.9-1_amd64.deb ...
Unpacking libwireshark18:amd64 (4.4.9-1) over (4.4.7-1+b1) ...
Preparing to unpack .../3-tshark_4.4.9-1_amd64.deb ...
Unpacking tshark (4.4.9-1) over (4.4.7-1+b1) ...
Preparing to unpack .../4-wireshark_4.4.9-1_amd64.deb ...
Unpacking wireshark (4.4.9-1) over (4.4.7-1+b1) ...
Preparing to unpack .../5-wireshark-common_4.4.9-1_amd64.deb ...
Unpacking wireshark-common (4.4.9-1) over (4.4.7-1+b1) ...
Preparing to unpack .../6-libwireshark-data_4.4.9-1_all.deb ...
Unpacking libwireshark-data (4.4.9-1) over (4.4.7-1) ...
Setting up libwireshark-data (4.4.9-1) ...
Setting up libwsutil16:amd64 (4.4.9-1) ...
Setting up libwiretap15:amd64 (4.4.9-1) ...
Setting up libwireshark18:amd64 (4.4.9-1) ...
Setting up wireshark-common (4.4.9-1) ...
Setting up wireshark (4.4.9-1) ...
Setting up tshark (4.4.9-1) ...
Processing triggers for mailcap (3.74) ...
Processing triggers for kali-menu (2025.3.2) ...
Processing triggers for desktop-file-utils (0.28-1) ...
Processing triggers for hicolor-icon-theme (0.18-2) ...
Processing triggers for libc-bin (2.41-12) ...
Processing triggers for man-db (2.13.1-1) ...
Processing triggers for shared-mime-info (2.4-5+b3) ...
```

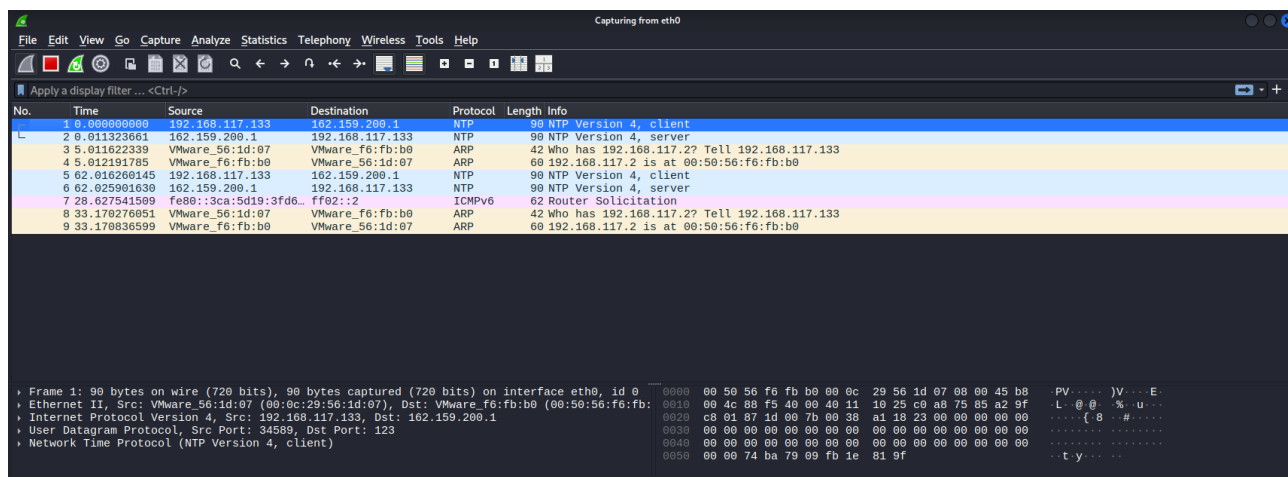
Step 2 :-

Start capturing the data packet's on Active Network.

Open the Wireshark –

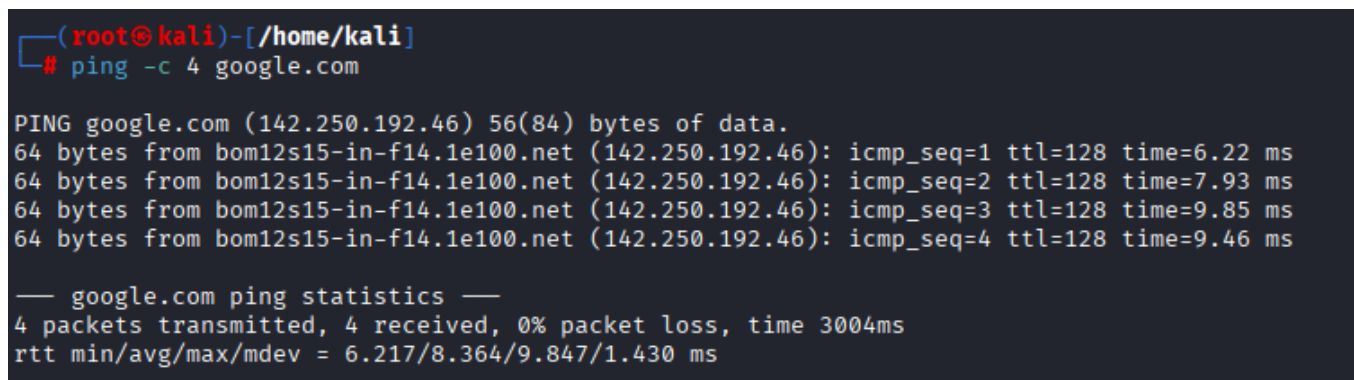


Capturing the data packets.

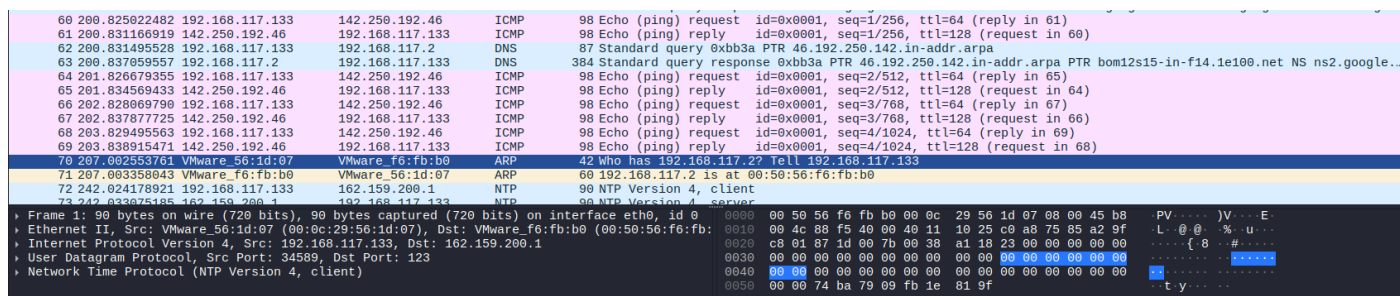


Step 3 :-

Generate the Network Traffics.

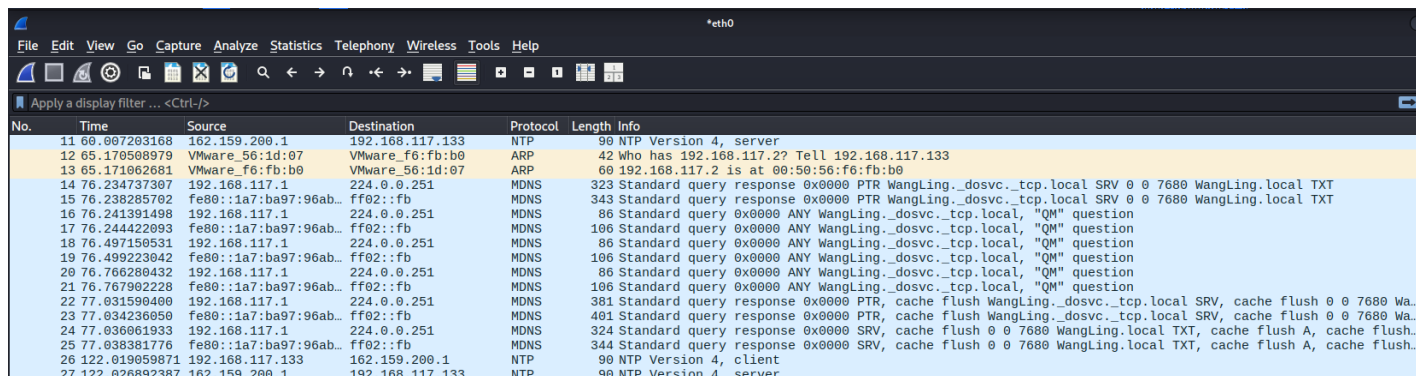


ICMP, DNS, ARP, NTP, etc packets are captured.



Step 4 :-

Let's stop the scan.

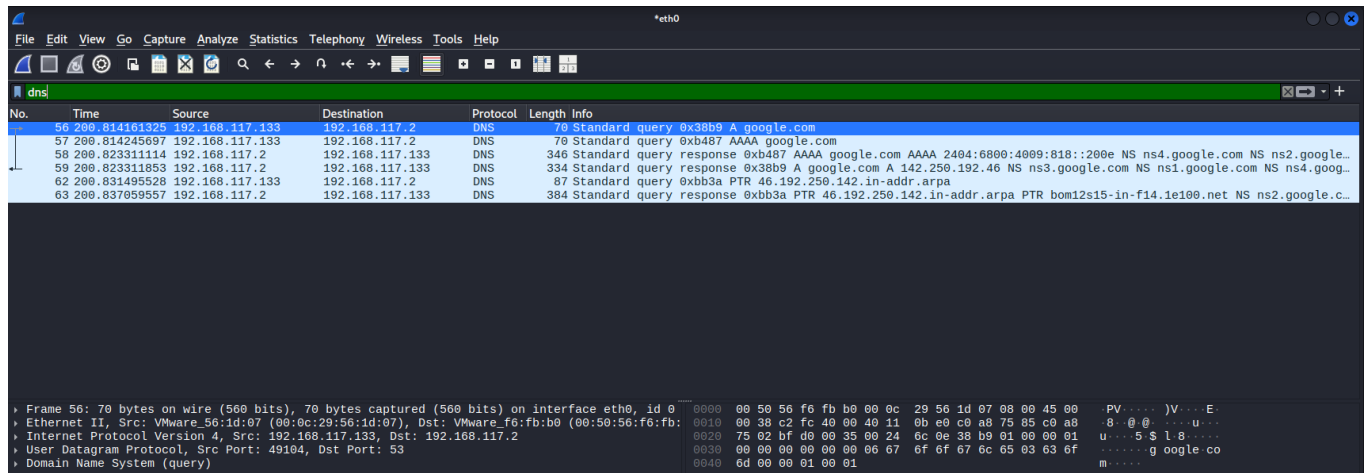


Step 5 :-

Filter the Captured Packets.

As, http, dns, tcp, icmp, etc.

Filtered for dns :-

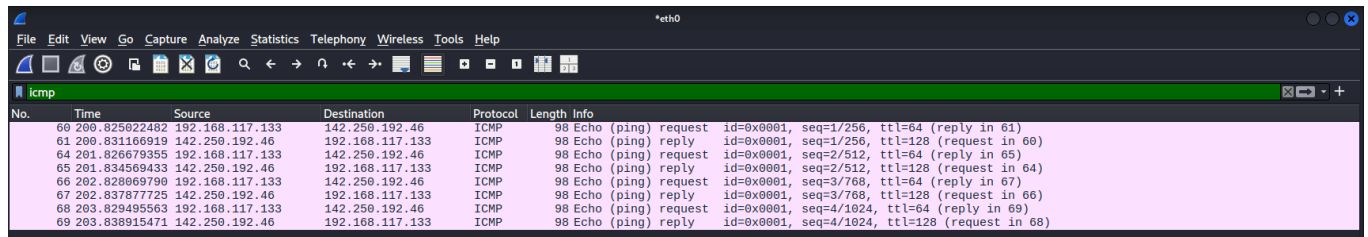


The screenshot shows the Wireshark interface with the filter 'dns' applied. The packet list displays several DNS queries and responses. The packet details pane shows the structure of a DNS query for 'google.com'.

No.	Time	Source	Destination	Protocol	Length	Info
56	200.814161325	192.168.117.133	192.168.117.2	DNS	70	Standard query 0x38b9 A google.com
57	200.814245697	192.168.117.133	192.168.117.2	DNS	70	Standard query 0xb487 AAAA google.com
58	200.823311114	192.168.117.2	192.168.117.133	DNS	346	Standard query response 0xb487 AAAA google.com AAAA 2404:6800:4009:818::200e NS ns4.google.com NS ns2.google...
59	200.823311853	192.168.117.2	192.168.117.133	DNS	334	Standard query response 0x38b9 A google.com A 142.250.192.46 NS ns3.google.com NS ns1.google.com NS ns4.goog...
62	200.831495528	192.168.117.133	192.168.117.2	DNS	87	Standard query 0xb33a PTR 46.192.250.142.in-addr.arpa PTR bom12s15-in-f14.1e100.net NS ns2.google.c...
63	200.837059557	192.168.117.2	192.168.117.133	DNS	384	Standard query response 0xb33a PTR 46.192.250.142.in-addr.arpa PTR bom12s15-in-f14.1e100.net NS ns2.google.c...

Frame 56: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface eth0, id 0
Ethernet II, Src: VMware_56:1d:07 (00:0c:29:56:1d:07), Dst: VMware_f6:fb:b0 (00:50:56:f6:fb:b0)
Internet Protocol Version 4, Src: 192.168.117.133, Dst: 192.168.117.2
User Datagram Protocol, Src Port: 49104, Dst Port: 53
Domain Name System (query)

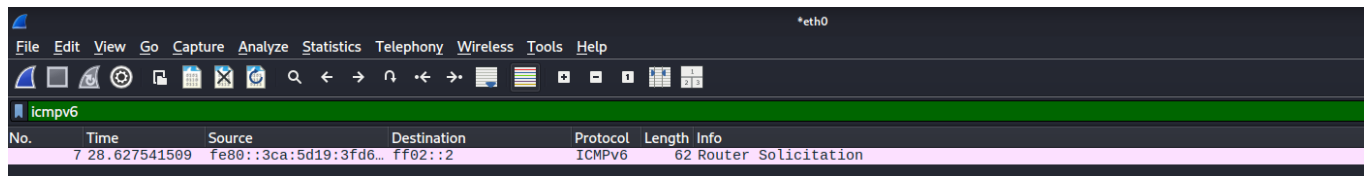
Filtered for icmp :-



The screenshot shows the Wireshark interface with the filter 'icmp' applied. The packet list displays several ICMP Echo (ping) requests and replies.

No.	Time	Source	Destination	Protocol	Length	Info
60	200.825922482	192.168.117.133	142.250.192.46	ICMP	98	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (reply in 61)
61	200.831166919	142.250.192.46	192.168.117.133	ICMP	98	Echo (ping) reply id=0x0001, seq=1/256, ttl=128 (request in 60)
64	201.826679355	192.168.117.133	142.250.192.46	ICMP	98	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (reply in 65)
65	201.834569433	142.250.192.46	192.168.117.133	ICMP	98	Echo (ping) reply id=0x0001, seq=2/512, ttl=128 (request in 64)
66	202.828697990	192.168.117.133	142.250.192.46	ICMP	98	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 67)
67	202.837877725	142.250.192.46	192.168.117.133	ICMP	98	Echo (ping) reply id=0x0001, seq=3/768, ttl=128 (request in 66)
68	203.829495563	192.168.117.133	142.250.192.46	ICMP	98	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (reply in 69)
69	203.838915471	142.250.192.46	192.168.117.133	ICMP	98	Echo (ping) reply id=0x0001, seq=4/1024, ttl=128 (request in 68)

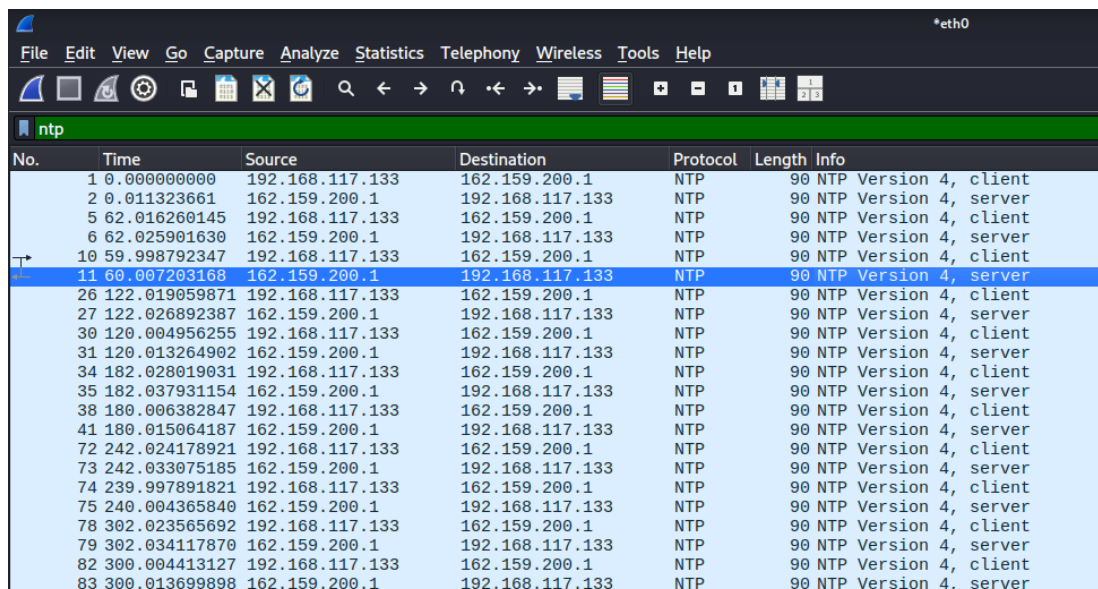
Filtered for icmp v6 :-



The screenshot shows the Wireshark interface with the filter 'icmpv6' applied. The packet list displays a single ICMPv6 Router Solicitation packet.

No.	Time	Source	Destination	Protocol	Length	Info
7	28.627541509	fe80::3ca:5d19:3fd6...	ff02::2	ICMPv6	62	Router Solicitation

Filtered for ntp :-



The screenshot shows the Wireshark interface with the filter 'ntp' applied. The packet list displays several NTP Version 4 packets, including client and server requests and responses.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.117.133	162.159.200.1	NTP	90	NTP Version 4, client
2	0.011323661	162.159.200.1	192.168.117.133	NTP	90	NTP Version 4, server
5	62.016260145	192.168.117.133	162.159.200.1	NTP	90	NTP Version 4, client
6	62.025901630	162.159.200.1	192.168.117.133	NTP	90	NTP Version 4, server
10	59.998792347	192.168.117.133	162.159.200.1	NTP	90	NTP Version 4, client
11	60.007203168	162.159.200.1	192.168.117.133	NTP	90	NTP Version 4, server
26	122.019059871	192.168.117.133	162.159.200.1	NTP	90	NTP Version 4, client
27	122.026892387	162.159.200.1	192.168.117.133	NTP	90	NTP Version 4, server
30	120.004956255	192.168.117.133	162.159.200.1	NTP	90	NTP Version 4, client
31	120.013264902	162.159.200.1	192.168.117.133	NTP	90	NTP Version 4, server
34	182.028019031	192.168.117.133	162.159.200.1	NTP	90	NTP Version 4, client
35	182.037931154	162.159.200.1	192.168.117.133	NTP	90	NTP Version 4, server
38	180.006382847	192.168.117.133	162.159.200.1	NTP	90	NTP Version 4, client
41	180.015064187	162.159.200.1	192.168.117.133	NTP	90	NTP Version 4, server
72	242.024178921	192.168.117.133	162.159.200.1	NTP	90	NTP Version 4, client
73	242.033075185	162.159.200.1	192.168.117.133	NTP	90	NTP Version 4, server
74	239.997891821	192.168.117.133	162.159.200.1	NTP	90	NTP Version 4, client
75	240.004365840	162.159.200.1	192.168.117.133	NTP	90	NTP Version 4, server
78	302.023565692	192.168.117.133	162.159.200.1	NTP	90	NTP Version 4, client
79	302.034117870	162.159.200.1	192.168.117.133	NTP	90	NTP Version 4, server
82	300.004413127	192.168.117.133	162.159.200.1	NTP	90	NTP Version 4, client
83	300.013699898	162.159.200.1	192.168.117.133	NTP	90	NTP Version 4, server

Step 6 :-

Identify atleast 3 different protocols.

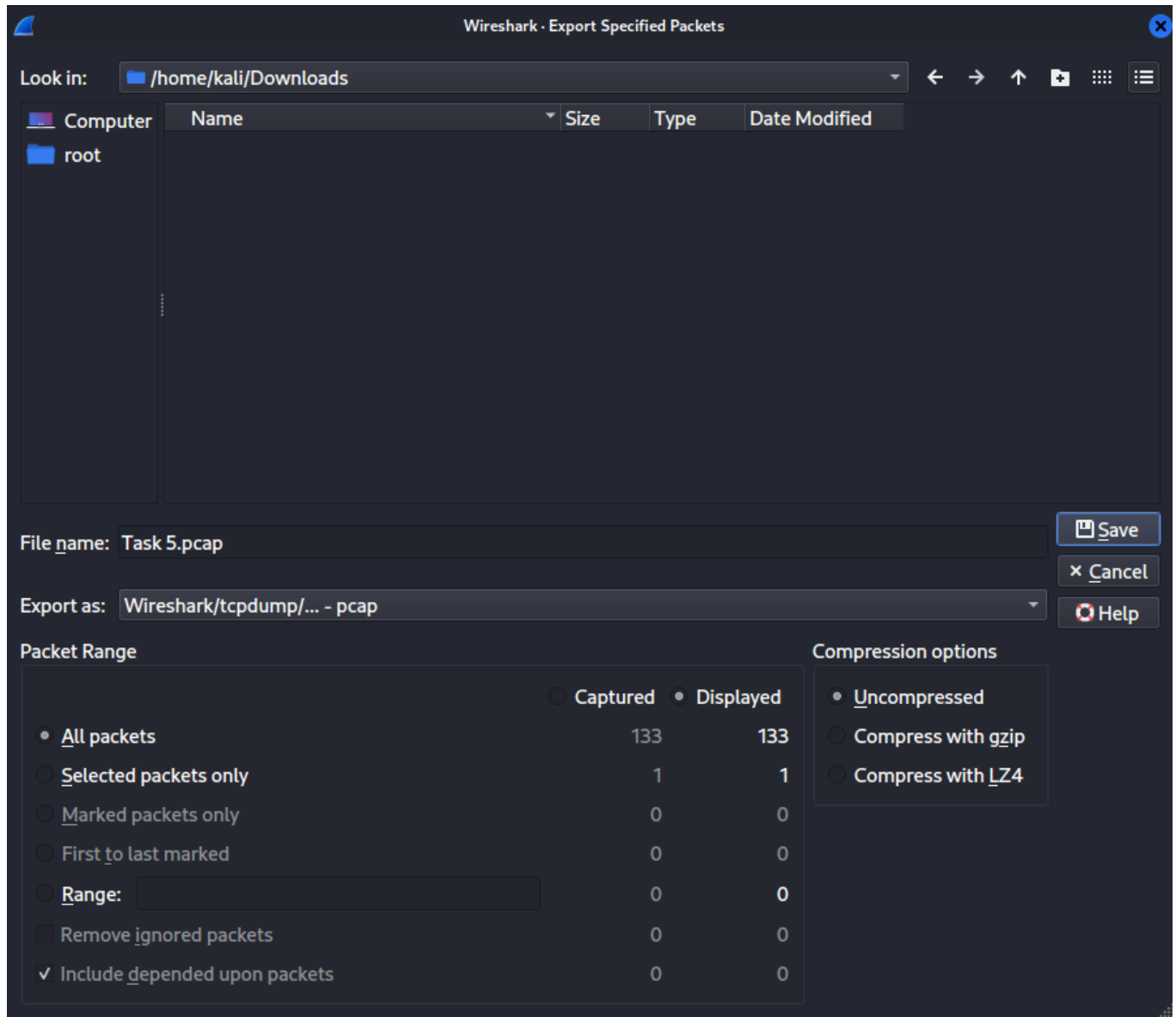
Protocol	Full Name	Layer (OSI Model)	Purpose / Function	Example Packet Details (Typical)	Notes for Wireshark
NTP	Network Time Protocol	Application Layer	Synchronizes the system clock with time servers (e.g., time.google.com)	Src: 192.168.1.5 Dst: 216.239.35.0 Protocol: UDP (port 123)	Filter: ntp or udp.port == 123
DNS	Domain Name System	Application Layer	Translates domain names (e.g., google.com) into IP addresses	Src: 192.168.1.5 Dst: 8.8.8.8 Protocol: UDP (port 53)	Filter: dns or udp.port == 53
ICMP	Internet Control Message Protocol (IPv4)	Network Layer	Used for diagnostics (e.g., ping) — echo request and reply messages	Src: 192.168.1.5 Dst: 142.250.64.78 (Google) Type: Echo (8) / Echo Reply (0)	Filter: icmp
ICMPv6	Internet Control Message Protocol for IPv6	Network Layer	Same as ICMP but for IPv6; used for neighbor discovery and ping6	Src: fe80::1 Dst: fe80::abcd:1234 Type: Echo Request (128) / Reply (129)	Filter: icmpv6
ARP	Address Resolution Protocol	Data Link Layer	Resolves IPv4 addresses to MAC addresses on local network	Src MAC: 08:00:27:aa:bb:cc Dst MAC: ff:ff:ff:ff:ff:ff Operation: Who has 192.168.1.1? Tell 192.168.1.5	Filter: arp

Step 7 :-

Export the capture as a .pcap file

File → Export Specified Packets...

And select for the PCAP file format.



Step 8 :-

Summarize the findings and packet details.

I captured network traffic for 60 seconds on interface wlan0 while browsing websites and pinging Google.

I identified multiple protocols including DNS, TCP, HTTP, ICMP, NTP, etc.

The DNS requests showed hostname lookups for domains visited, HTTP packets revealed GET requests for web pages, and ICMP packets were generated by the ping command.