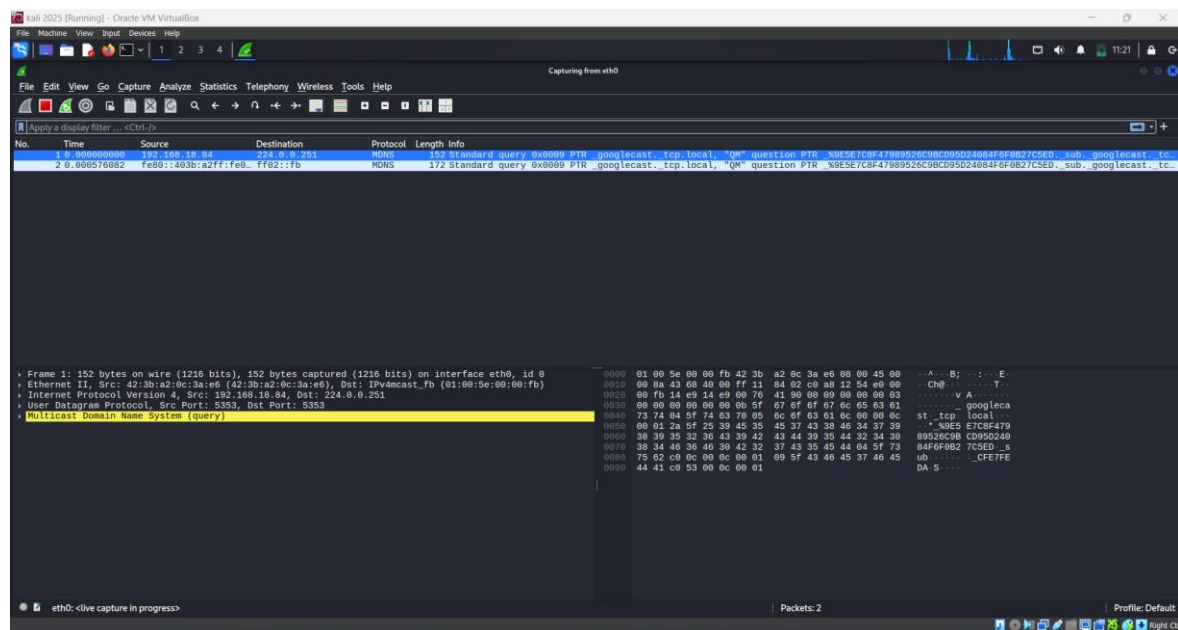


Wireshark Packet Capture Report

This report documents the process of capturing and analyzing network traffic using Wireshark on Kali Linux. Each step includes a screenshot and explanation for better understanding.

Step 1: Opening Wireshark

In this step, Wireshark is launched on Kali Linux. The main interface shows the available network interfaces that can be monitored.



After that I ping to ping 8.8.8.8 (Google)

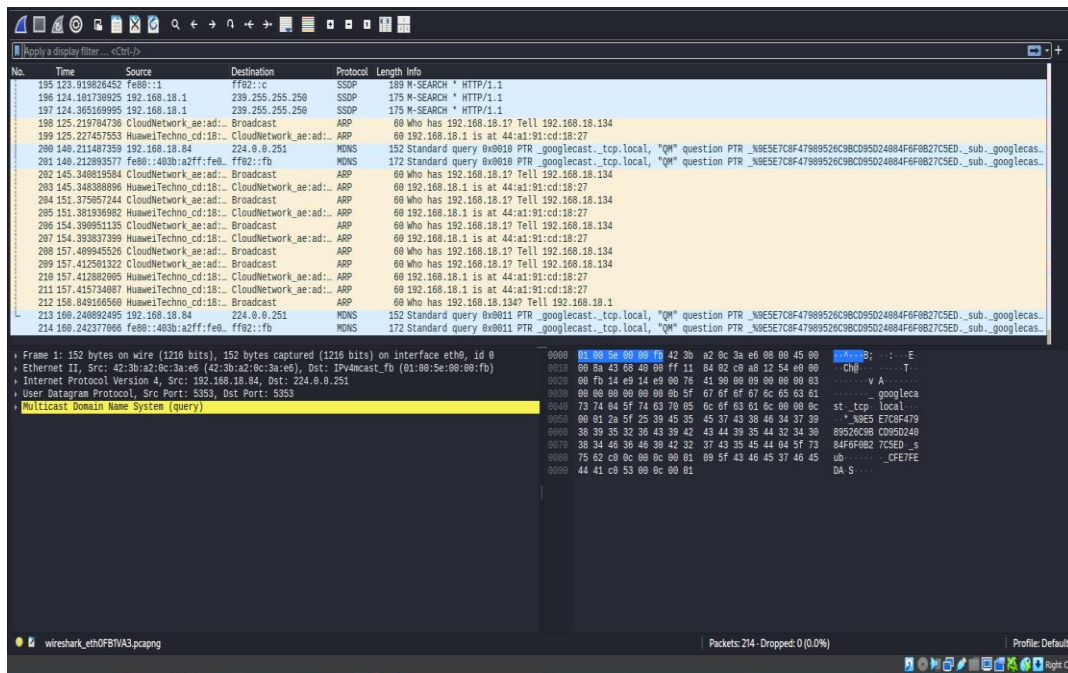
```

(root@KALI)-[~]
# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=34.5 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=24.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=118 time=72.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=118 time=24.6 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=118 time=21.0 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=118 time=38.6 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=118 time=41.8 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=118 time=25.9 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=118 time=23.2 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=118 time=26.4 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=118 time=25.7 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=118 time=19.9 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=118 time=25.7 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=118 time=21.2 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=118 time=23.7 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=118 time=28.3 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=118 time=20.5 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=118 time=21.0 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=118 time=23.2 ms
64 bytes from 8.8.8.8: icmp_seq=20 ttl=118 time=31.2 ms
64 bytes from 8.8.8.8: icmp_seq=21 ttl=118 time=20.4 ms
64 bytes from 8.8.8.8: icmp_seq=22 ttl=118 time=21.4 ms
64 bytes from 8.8.8.8: icmp_seq=23 ttl=118 time=22.5 ms
64 bytes from 8.8.8.8: icmp_seq=24 ttl=118 time=25.0 ms
64 bytes from 8.8.8.8: icmp_seq=25 ttl=118 time=30.9 ms
64 bytes from 8.8.8.8: icmp_seq=26 ttl=118 time=32.6 ms
64 bytes from 8.8.8.8: icmp_seq=27 ttl=118 time=22.6 ms
64 bytes from 8.8.8.8: icmp_seq=28 ttl=118 time=21.4 ms
64 bytes from 8.8.8.8: icmp_seq=29 ttl=118 time=21.7 ms
64 bytes from 8.8.8.8: icmp_seq=30 ttl=118 time=27.8 ms
64 bytes from 8.8.8.8: icmp_seq=31 ttl=118 time=50.8 ms
64 bytes from 8.8.8.8: icmp_seq=32 ttl=118 time=21.1 ms
64 bytes from 8.8.8.8: icmp_seq=33 ttl=118 time=28.3 ms
64 bytes from 8.8.8.8: icmp_seq=34 ttl=118 time=32.3 ms
64 bytes from 8.8.8.8: icmp_seq=35 ttl=118 time=39.0 ms
64 bytes from 8.8.8.8: icmp_seq=36 ttl=118 time=31.0 ms
64 bytes from 8.8.8.8: icmp_seq=37 ttl=118 time=22.3 ms
64 bytes from 8.8.8.8: icmp_seq=38 ttl=118 time=21.4 ms
64 bytes from 8.8.8.8: icmp_seq=39 ttl=118 time=26.5 ms
64 bytes from 8.8.8.8: icmp_seq=40 ttl=118 time=20.9 ms
64 bytes from 8.8.8.8: icmp_seq=41 ttl=118 time=21.9 ms
64 bytes from 8.8.8.8: icmp_seq=42 ttl=118 time=23.5 ms
64 bytes from 8.8.8.8: icmp_seq=43 ttl=118 time=28.0 ms
64 bytes from 8.8.8.8: icmp_seq=44 ttl=118 time=25.2 ms
64 bytes from 8.8.8.8: icmp_seq=45 ttl=118 time=36.8 ms
64 bytes from 8.8.8.8: icmp_seq=46 ttl=118 time=27.2 ms
64 bytes from 8.8.8.8: icmp_seq=47 ttl=118 time=20.1 ms
64 bytes from 8.8.8.8: icmp_seq=48 ttl=118 time=24.3 ms
64 bytes from 8.8.8.8: icmp_seq=49 ttl=118 time=20.8 ms
^C
— 8.8.8.8 ping statistics —
49 packets transmitted, 49 received, 0% packet loss, time 48114ms
rtt min/avg/max/mdev = 19.913/27.360/72.557/9.142 ms

```

Step 2: Starting Packet Capture

The active network interface (such as eth0 or wlan0) is selected, and the 'Start Capturing Packets' button is clicked to begin capturing live network traffic.



Step 3: Filtering Traffic by Protocol

After capturing data, filters such as 'HTTP', 'DNS', or ICMP' are applied to view only relevant packets. This helps in focusing on specific network activities.

No.	Time	Source	Destination	Protocol	Length	Info
193	123.91826452	fe80::1	ff02::c	SSDP	189	M-SEARCH * HTTP/1.1
196	124.181738925	192.168.18.1	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
197	124.365169995	192.168.18.1	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
198	125.219784736	CloudNetwork_ae:ad...	Broadcast	ARP	60	Who has 192.168.18.1? Tell 192.168.18.134
199	125.227457553	HuaweiTechno_cd:18...	CloudNetwork_ae:ad...	ARP	60	192.168.18.1 is at 44:a1:91:cd:18:27
200	140.211487359	192.168.18.84	224.0.0.251	MDNS	152	Standard query 0x0010 PTR googlecast._tcp.local, "QM" question PTR _N9E5E7C8F47889526C9BCD95D24884F6F0B27C5ED._sub._googlecas...
201	140.211487359	192.168.18.84	224.0.0.251	MDNS	172	Standard query 0x0010 PTR googlecast._tcp.local, "QM" question PTR _N9E5E7C8F47889526C9BCD95D24884F6F0B27C5ED._sub._googlecas...
202	145.348819584	CloudNetwork_ae:ad...	Broadcast	ARP	60	Who has 192.168.18.1? Tell 192.168.18.134
203	145.348888896	HuaweiTechno_cd:18...	CloudNetwork_ae:ad...	ARP	60	192.168.18.1 is at 44:a1:91:cd:18:27
204	151.375857244	CloudNetwork_ae:ad...	Broadcast	ARP	60	Who has 192.168.18.1? Tell 192.168.18.134
205	151.381369682	HuaweiTechno_cd:18...	CloudNetwork_ae:ad...	ARP	60	192.168.18.1 is at 44:a1:91:cd:18:27
206	154.399951135	CloudNetwork_ae:ad...	Broadcast	ARP	60	Who has 192.168.18.1? Tell 192.168.18.134
207	154.399837399	HuaweiTechno_cd:18...	CloudNetwork_ae:ad...	ARP	60	192.168.18.1 is at 44:a1:91:cd:18:27
208	157.409945526	CloudNetwork_ae:ad...	Broadcast	ARP	60	Who has 192.168.18.1? Tell 192.168.18.134
209	157.412591322	CloudNetwork_ae:ad...	Broadcast	ARP	60	Who has 192.168.18.1? Tell 192.168.18.134
210	157.412882865	HuaweiTechno_cd:18...	CloudNetwork_ae:ad...	ARP	60	192.168.18.1 is at 44:a1:91:cd:18:27
211	157.415734887	HuaweiTechno_cd:18...	CloudNetwork_ae:ad...	ARP	60	192.168.18.1 is at 44:a1:91:cd:18:27
212	158.849166568	HuaweiTechno_cd:18...	Broadcast	ARP	60	Who has 192.168.18.134? Tell 192.168.18.1
213	160.248892495	192.168.18.84	224.0.0.251	MDNS	152	Standard query 0x0011 PTR googlecast._tcp.local, "QM" question PTR _N9E5E7C8F47889526C9BCD95D24884F6F0B27C5ED._sub._googlecas...
214	160.242377866	fe80::493b:a2ff:fe...	ff02::fb	MDNS	172	Standard query 0x0011 PTR googlecast._tcp.local, "QM" question PTR _N9E5E7C8F47889526C9BCD95D24884F6F0B27C5ED._sub._googlecas...

Step 4: Viewing Packet Details

A selected packet is expanded to show detailed protocol layers (Ethernet, IP, TCP/UDP, etc.) along with data. This allows for in-depth analysis of communication between devices.

15	6.594130934	CloudNetwork_ae:ad1..	ARP	62	Who has 192.168.18.1? Tell 192.168.18.134
43	20.040858312	PCSSystemec_3c:2a:..	ARP	44	Who has 192.168.18.1? Tell 192.168.18.171
44	20.052109020	HuaweiTechno_cd:18:..	ARP	62	192.168.18.1 is at 44:a1:91:cd:18:27
58	26.707863465	CloudNetwork_ae:ad1..	ARP	62	Who has 192.168.18.1? Tell 192.168.18.134
69	31.734673621	CloudNetwork_ae:ad1..	ARP	62	Who has 192.168.18.1? Tell 192.168.18.134
74	43.499493826	42:3b:a2:0c:3a:e6	ARP	62	Who has 192.168.18.1? Tell 192.168.18.84
75	48.031965668	CloudNetwork_ae:ad1..	ARP	62	Who has 192.168.18.1? Tell 192.168.18.134
76	52.051395625	CloudNetwork_ae:ad1..	ARP	62	Who has 192.168.18.1? Tell 192.168.18.134
77	56.071902647	CloudNetwork_ae:ad1..	ARP	62	Who has 192.168.18.1? Tell 192.168.18.134
78	72.965168755	CloudNetwork_ae:ad1..	ARP	62	Who has 192.168.18.1? Tell 192.168.18.134
79	85.031118401	CloudNetwork_ae:ad1..	ARP	62	Who has 192.168.18.1? Tell 192.168.18.134
80	89.053448777	CloudNetwork_ae:ad1..	ARP	62	Who has 192.168.18.1? Tell 192.168.18.134

Conclusion

The above steps demonstrate how to use Wireshark for basic packet capturing and protocol analysis. Through this exercise, we identified different types of traffic and learned to filter and inspect packets in detail.