TASK 5

CAPTURE AND ANALYSE NETWORK TRAFFIC USING WIRESHARK

By - Sharine R Prakash

Objective:

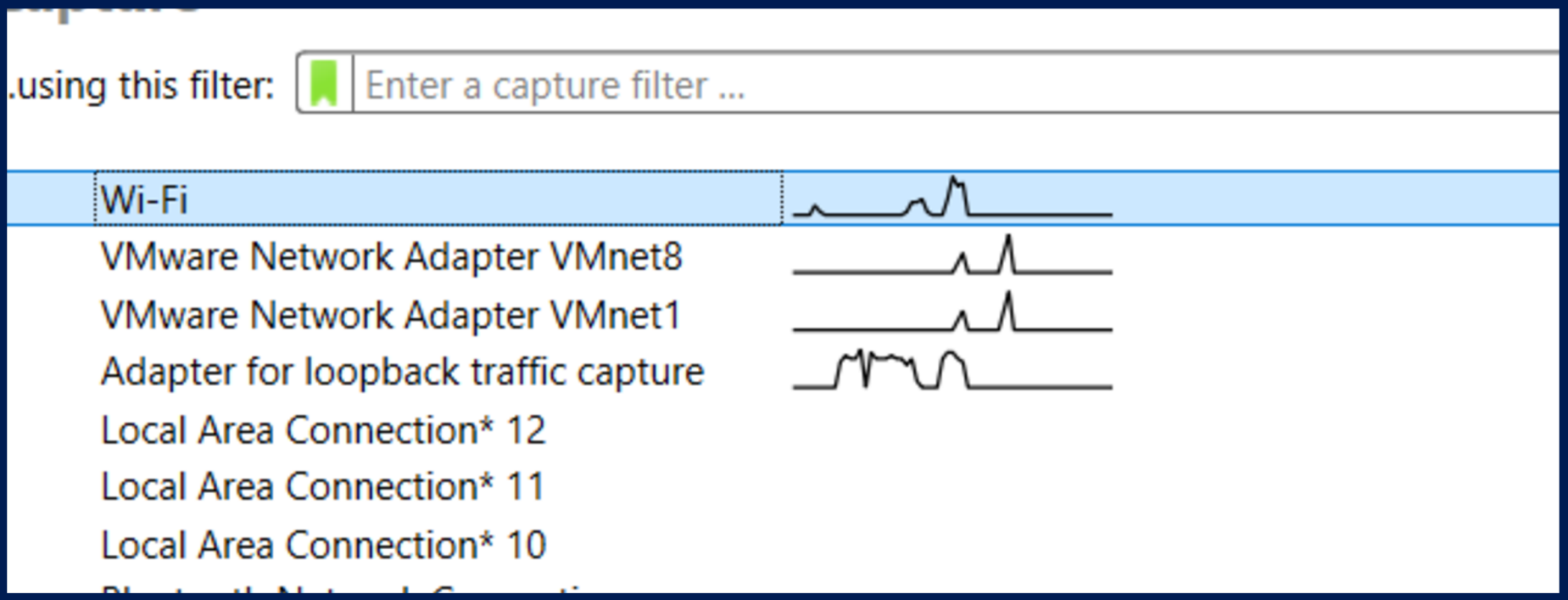
To capture live network packets using Wireshark and analyze protocols such as DNS, HTTP, TCP, and ICMPv6. The purpose was to understand how data is transmitted between devices and how different network protocols operate in real-time communication.

Tools:

- Wireshark – For capturing and analyzing packets
- Web Browser (Chrome) – To generate internet traffic
- Operating System – Windows 11

Steps Performed:

- Opened Wireshark and selected the active Wi-Fi interface.

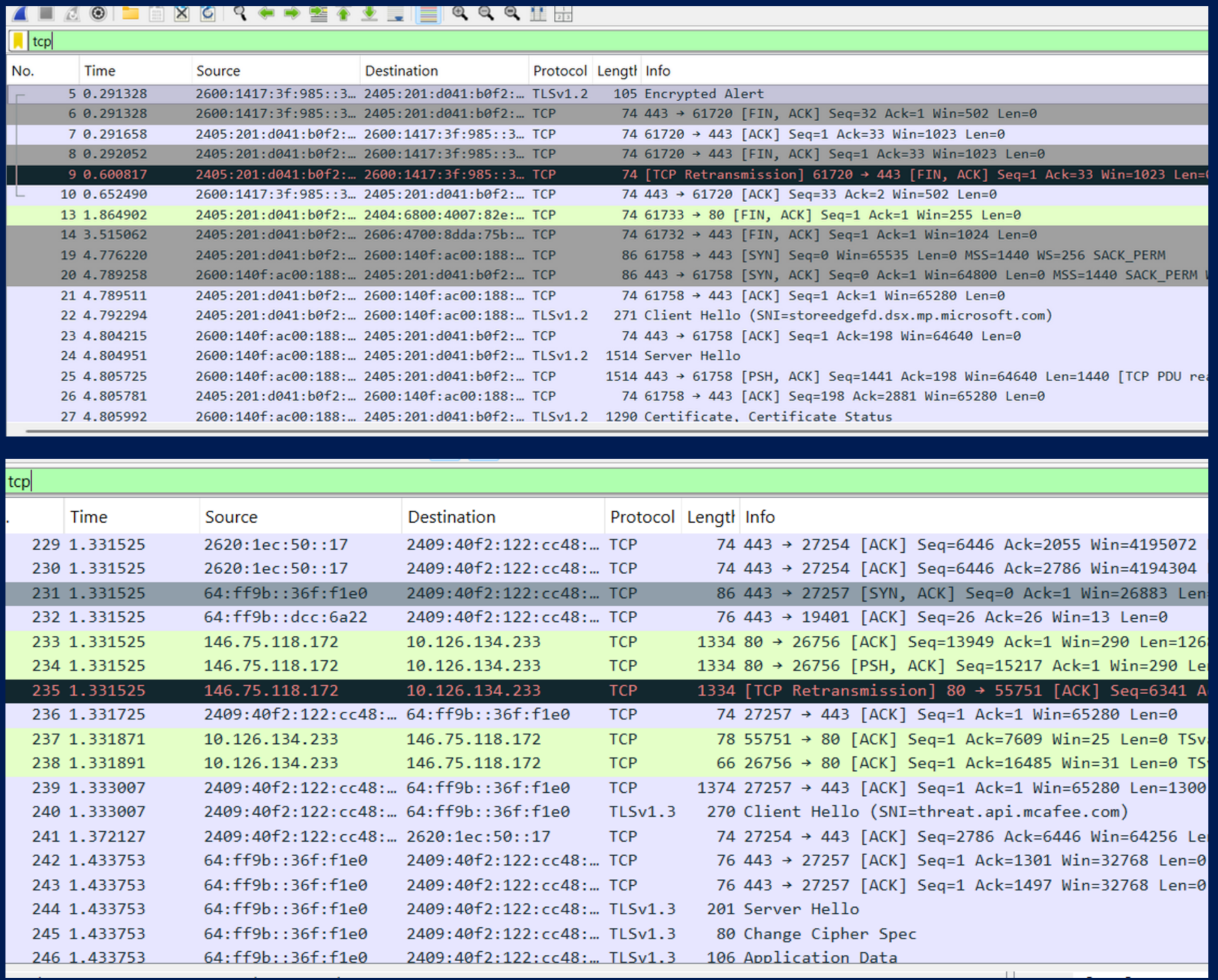


- Started packet capture and browsed multiple websites.

https://www.google.com
https://api.github.com (in browser or curl)
https://www.microsoft.com
https://www.amazon.in
https://skype.com

- **Applied protocol filters: dns, http, tcp, and icmpv6.**

TCP (Transmission Control Protocol)



tcp

No.	Time	Source	Destination	Protocol	Length	Info
5	0.291328	2600:1417:3f:985::3…	2405:201:d041:b0f2:…	TLSv1.2	105	Encrypted Alert
6	0.291328	2600:1417:3f:985::3…	2405:201:d041:b0f2:…	TCP	74	443 → 61720 [FIN, ACK] Seq=32 Ack=1 Win=502 Len=0
7	0.291658	2405:201:d041:b0f2:…	2600:1417:3f:985::3…	TCP	74	61720 → 443 [ACK] Seq=1 Ack=33 Win=1023 Len=0
8	0.292052	2405:201:d041:b0f2:…	2600:1417:3f:985::3…	TCP	74	61720 → 443 [FIN, ACK] Seq=1 Ack=33 Win=1023 Len=0
9	0.600817	2405:201:d041:b0f2:…	2600:1417:3f:985::3…	TCP	74	[TCP Retransmission] 61720 → 443 [FIN, ACK] Seq=1 Ack=33 Win=1023 Len=(
10	0.652490	2600:1417:3f:985::3…	2405:201:d041:b0f2:…	TCP	74	443 → 61720 [ACK] Seq=33 Ack=2 Win=502 Len=0
13	1.864902	2405:201:d041:b0f2:…	2404:6800:4007:82e:…	TCP	74	61733 → 80 [FIN, ACK] Seq=1 Ack=1 Win=255 Len=0
14	3.515062	2405:201:d041:b0f2:…	2606:4700:8dda:75b:…	TCP	74	61732 → 443 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
19	4.776220	2405:201:d041:b0f2:…	2600:140f:ac00:188:…	TCP	86	61758 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1440 WS=256 SACK_PERM
20	4.789258	2600:140f:ac00:188:…	2405:201:d041:b0f2:…	TCP	86	443 → 61758 [SYN, ACK] Seq=0 Ack=1 Win=64800 Len=0 MSS=1440 SACK_PERM
21	4.789511	2405:201:d041:b0f2:…	2600:140f:ac00:188:…	TCP	74	61758 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=0
22	4.792294	2405:201:d041:b0f2:…	2600:140f:ac00:188:…	TLSv1.2	271	Client Hello (SNI=storeedgefd.dsx.mp.microsoft.com)
23	4.804215	2600:140f:ac00:188:…	2405:201:d041:b0f2:…	TCP	74	443 → 61758 [ACK] Seq=1 Ack=198 Win=64640 Len=0
24	4.804951	2600:140f:ac00:188:…	2405:201:d041:b0f2:…	TLSv1.2	1514	Server Hello
25	4.805725	2600:140f:ac00:188:…	2405:201:d041:b0f2:…	TCP	1514	443 → 61758 [PSH, ACK] Seq=1441 Ack=198 Win=64640 Len=1440 [TCP PDU re
26	4.805781	2405:201:d041:b0f2:…	2600:140f:ac00:188:…	TCP	74	61758 → 443 [ACK] Seq=198 Ack=2881 Win=65280 Len=0
27	4.805992	2600:140f:ac00:188:…	2405:201:d041:b0f2:…	TLSv1.2	1290	Certificate, Certificate Status

tcp

.	Time	Source	Destination	Protocol	Length	Info
229	1.331525	2620:1ec:50::17	2409:40f2:122:cc48:…	TCP	74	443 → 27254 [ACK] Seq=6446 Ack=2055 Win=4195072
230	1.331525	2620:1ec:50::17	2409:40f2:122:cc48:…	TCP	74	443 → 27254 [ACK] Seq=6446 Ack=2786 Win=4194304
231	1.331525	64:ff9b::36f:f1e0	2409:40f2:122:cc48:…	TCP	86	443 → 27257 [SYN, ACK] Seq=0 Ack=1 Win=26883 Len
232	1.331525	64:ff9b::dcc:6a22	2409:40f2:122:cc48:…	TCP	76	443 → 19401 [ACK] Seq=26 Ack=26 Win=13 Len=0
233	1.331525	146.75.118.172	10.126.134.233	TCP	1334	80 → 26756 [ACK] Seq=13949 Ack=1 Win=290 Len=126
234	1.331525	146.75.118.172	10.126.134.233	TCP	1334	80 → 26756 [PSH, ACK] Seq=15217 Ack=1 Win=290 Le
235	1.331525	146.75.118.172	10.126.134.233	TCP	1334	[TCP Retransmission] 80 → 55751 [ACK] Seq=6341 A
236	1.331725	2409:40f2:122:cc48:…	64:ff9b::36f:f1e0	TCP	74	27257 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=0
237	1.331871	10.126.134.233	146.75.118.172	TCP	78	55751 → 80 [ACK] Seq=1 Ack=7609 Win=25 Len=0 TSv
238	1.331891	10.126.134.233	146.75.118.172	TCP	66	26756 → 80 [ACK] Seq=1 Ack=16485 Win=31 Len=0 TS
239	1.333007	2409:40f2:122:cc48:…	64:ff9b::36f:f1e0	TCP	1374	27257 → 443 [ACK] Seq=1 Ack=1 Win=65280 Len=1300
240	1.333007	2409:40f2:122:cc48:…	64:ff9b::36f:f1e0	TLSv1.3	270	Client Hello (SNI=threat.api.mcafee.com)
241	1.372127	2409:40f2:122:cc48:…	2620:1ec:50::17	TCP	74	27254 → 443 [ACK] Seq=2786 Ack=6446 Win=64256 Le
242	1.433753	64:ff9b::36f:f1e0	2409:40f2:122:cc48:…	TCP	76	443 → 27257 [ACK] Seq=1 Ack=1301 Win=32768 Len=0
243	1.433753	64:ff9b::36f:f1e0	2409:40f2:122:cc48:…	TCP	76	443 → 27257 [ACK] Seq=1 Ack=1497 Win=32768 Len=0
244	1.433753	64:ff9b::36f:f1e0	2409:40f2:122:cc48:…	TLSv1.3	201	Server Hello
245	1.433753	64:ff9b::36f:f1e0	2409:40f2:122:cc48:…	TLSv1.3	80	Change Cipher Spec
246	1.433753	64:ff9b::36f:f1e0	2409:40f2:122:cc48:…	TLSv1.3	106	Application Data

- TCP is a connection-oriented protocol that ensures reliable data transfer between devices.
- It uses a process called the "Three-Way Handshake" (SYN, SYN-ACK, ACK) to establish a secure and ordered connection.
- TCP segments are retransmitted if lost, guaranteeing complete delivery.
- In your capture, TCP packets were observed supporting HTTP and DNS traffic, ensuring reliability in data communication.
- From a security perspective, monitoring TCP helps identify unauthorized open ports or unusual connection attempts, which may indicate scanning or intrusion activities.

HTTP (HyperText Transfer Protocol)

http

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	146.75.122.172	10.126.134.233	HTTP	1334	Continuation
2	0.000000	146.75.122.172	10.126.134.233	HTTP	1334	Continuation
9	0.002933	146.75.122.172	10.126.134.233	HTTP	1334	Continuation
10	0.002933	146.75.122.172	10.126.134.233	HTTP	1334	Continuation
11	0.002933	146.75.122.172	10.126.134.233	HTTP	1334	Continuation
12	0.002933	146.75.122.172	10.126.134.233	HTTP	1334	Continuation
13	0.002933	146.75.122.172	10.126.134.233	HTTP	1334	Continuation
14	0.002933	146.75.122.172	10.126.134.233	HTTP	638	Continuation
17	0.111911	2a04:4e42:8e::684	2409:40f2:122:cc48:…	HTTP	1374	Continuation
18	0.111911	2a04:4e42:8e::684	2409:40f2:122:cc48:…	HTTP	1374	Continuation
19	0.111911	2a04:4e42:8e::684	2409:40f2:122:cc48:…	HTTP	1374	Continuation
20	0.111911	2a04:4e42:8e::684	2409:40f2:122:cc48:…	HTTP	1086	Continuation
21	0.111911	2a04:4e42:8e::684	2409:40f2:122:cc48:…	HTTP	1374	Continuation
22	0.111911	2a04:4e42:8e::684	2409:40f2:122:cc48:…	HTTP	1374	Continuation
24	0.111911	2a04:4e42:8e::684	2409:40f2:122:cc48:…	HTTP	1374	Continuation
25	0.111911	2a04:4e42:8e::684	2409:40f2:122:cc48:…	HTTP	1374	Continuation

- HTTP is an application-layer protocol used for transferring web content such as text, images, and multimedia.
- It works on top of TCP, typically using port 80, and allows browsers to request web pages from servers.
- In the capture, HTTP traffic from neverssl.com displayed GET and response packets showing unencrypted data.
- This demonstrates how older websites transmit data in plain text, which can be intercepted easily.
- In cybersecurity, inspecting HTTP traffic helps identify insecure web sessions or potential data leakage in unencrypted form.

DNS (Domain Name System)

dns

No.	Time	Source	Destination	Protocol	Length	Info
103	0.802622	10.126.134.233	10.126.134.246	DNS	81	Standard query 0xe549 A config.edge.skype.com
104	0.803625	10.126.134.233	10.126.134.246	DNS	81	Standard query 0x24ba AAAA config.edge.skype.c
134	0.888044	10.126.134.246	10.126.134.233	DNS	244	Standard query response 0xe549 A config.edge.s
135	0.888044	10.126.134.246	10.126.134.233	DNS	256	Standard query response 0x24ba AAAA config.edg
188	1.175357	10.126.134.233	10.126.134.246	DNS	81	Standard query 0x9a7e A threat.api.mcafee.com
189	1.175647	10.126.134.233	10.126.134.246	DNS	81	Standard query 0xece5 AAAA threat.api.mcafee.c
219	1.246452	10.126.134.246	10.126.134.233	DNS	148	Standard query response 0x9a7e A threat.api.mc
221	1.246701	10.126.134.246	10.126.134.233	DNS	172	Standard query response 0xece5 AAAA threat.api
1743	7.372790	2409:40f2:122:cc48:…	2409:40f2:122:cc48:…	DNS	106	Standard query 0x9c99 AAAA api.github.com
1745	7.373069	2409:40f2:122:cc48:…	2409:40f2:122:cc48:…	DNS	106	Standard query 0x8754 HTTPS api.github.com
1750	7.373247	2409:40f2:122:cc48:…	2409:40f2:122:cc48:…	DNS	106	Standard query 0x0c7c A api.github.com
1753	7.373453	2409:40f2:122:cc48:…	2409:40f2:122:cc48:…	DNS	112	Standard query 0xbec5 AAAA collector.github.co
1755	7.373782	2409:40f2:122:cc48:…	2409:40f2:122:cc48:…	DNS	112	Standard query 0x40c9 HTTPS collector.github.c
1757	7.373983	2409:40f2:122:cc48:…	2409:40f2:122:cc48:…	DNS	112	Standard query 0x4f2d A collector.github.com
1785	7.402629	2409:40f2:122:cc48:…	2409:40f2:122:cc48:…	DNS	104	Standard query 0x3f1b A srtb.msn.com
1787	7.402892	2409:40f2:122:cc48:…	2409:40f2:122:cc48:…	DNS	104	Standard query 0x561d AAAA srtb.msn.com
1789	7.403044	2409:40f2:122:cc48:…	2409:40f2:122:cc48:…	DNS	104	Standard query 0x1bc2 HTTPS srtb.msn.com
1856	7.477827	2409:40f2:122:cc48:…	2409:40f2:122:cc48:…	DNS	101	Standard query 0x18cf HTTPS c.msn.com

- The DNS protocol is used to translate human-readable domain names (like www.google.com) into IP addresses that computers use to identify each other on the network.
- When you visit a website, your computer first sends a DNS query to find the corresponding IP address.
- In the captured traffic, DNS packets show queries for domains such as config.edge.skype.com and api.github.com.
- DNS mainly operates over UDP port 53 but can also use TCP for large queries or zone transfers.
- In cybersecurity, analyzing DNS helps detect suspicious connections, phishing domains, or malware contacting command-and-control servers.

ICMPv6 (Internet Control Message Protocol for IPv6)

icmpv6

o.	Time	Source	Destination	Protocol	Length	Info
11	1.703913	fe80::d305:170d:e76…	fe80::b6a7:c6ff:fe8…	ICMPv6	86	Neighbor Solicitation for fe80::b6a7:c6ff:fe80:cce2 from
12	1.707111	fe80::b6a7:c6ff:fe8…	fe80::d305:170d:e76…	ICMPv6	78	Neighbor Advertisement fe80::b6a7:c6ff:fe80:cce2 (rtr, s
79	6.715586	fe80::b6a7:c6ff:fe8…	fe80::d305:170d:e76…	ICMPv6	86	Neighbor Solicitation for fe80::d305:170d:e76e:3f6c from
80	6.715801	fe80::d305:170d:e76…	fe80::b6a7:c6ff:fe8…	ICMPv6	86	Neighbor Advertisement fe80::d305:170d:e76e:3f6c (sol, c
107	9.808449	fe80::b6a7:c6ff:fe8…	2405:201:d041:b0f2:…	ICMPv6	86	Neighbor Solicitation for 2405:201:d041:b0f2:d0ae:26ad:a
108	9.808535	2405:201:d041:b0f2:…	fe80::b6a7:c6ff:fe8…	ICMPv6	86	Neighbor Advertisement 2405:201:d041:b0f2:d0ae:26ad:a66:
109	10.289916	fe80::b6a7:c6ff:fe8…	ff02::1	ICMPv6	142	Router Advertisement from b4:a7:c6:80:cc:e2
110	10.289916	fe80::b6a7:c6ff:fe8…	ff02::1	ICMPv6	142	Router Advertisement from b4:a7:c6:80:cc:e2
111	10.289916	fe80::b6a7:c6ff:fe8…	ff02::1	ICMPv6	142	Router Advertisement from b4:a7:c6:80:cc:e2
197	22.986039	fe80::b6a7:c6ff:fe8…	ff02::1	ICMPv6	142	Router Advertisement from b4:a7:c6:80:cc:e2
198	22.986039	fe80::b6a7:c6ff:fe8…	ff02::1	ICMPv6	142	Router Advertisement from b4:a7:c6:80:cc:e2
199	22.988282	fe80::b6a7:c6ff:fe8…	ff02::1	ICMPv6	142	Router Advertisement from b4:a7:c6:80:cc:e2
243	35.697456	2405:201:d041:b0f2:…	2405:201:d041:b0f2:…	ICMPv6	86	Neighbor Solicitation for 2405:201:d041:b0f2::c0a8:1d01
244	35.700038	2405:201:d041:b0f2:…	2405:201:d041:b0f2:…	ICMPv6	78	Neighbor Advertisement 2405:201:d041:b0f2::c0a8:1d01 (rt
324	62.701736	fe80::d305:170d:e76…	fe80::b6a7:c6ff:fe8…	ICMPv6	86	Neighbor Solicitation for fe80::b6a7:c6ff:fe80:cce2 from
325	62.705164	fe80::b6a7:c6ff:fe8…	fe80::d305:170d:e76…	ICMPv6	78	Neighbor Advertisement fe80::b6a7:c6ff:fe80:cce2 (rtr, s
335	67.707676	fe80::b6a7:c6ff:fe8…	fe80::d305:170d:e76…	ICMPv6	86	Neighbor Solicitation for fe80::d305:170d:e76e:3f6c from

- ICMPv6 is used by IPv6 networks for diagnostics, error reporting, and neighbor discovery.
- It helps devices communicate status information, such as unreachable destinations or packet delivery failures.
- In your capture, ICMPv6 packets indicated system-level communication between your computer and other IPv6-enabled devices on the network.
- Unlike TCP or HTTP, ICMPv6 does not carry user data — instead, it supports the health and functionality of the network.
- From a cybersecurity perspective, attackers sometimes use ICMPv6 to map network topology or detect live hosts (ping sweeps), making its analysis crucial.

Results

Successfully captured and analyzed multiple protocols including DNS, HTTP, TCP, and ICMPv6.
The DNS analysis revealed domain lookups, HTTP showed plain text web requests, TCP displayed reliable transmission, and ICMPv6 handled IPv6 communication.

Files and References:

📁 Files Included

📄 traffic_capture.pcap – Packet capture file

📘 README.md – Documentation summary

🔗 Reference:

Wireshark Official Website – https://www.wireshark.org/