NETWORK TRAFFIC CAPTURE AND ANALYSIS REPORT

Cyber Security Internship – Task 5 Elevate Labs | Ministry of MSME, Government of India

Submitted by: Tayyeba Ali **Date:** 01/07/2025

1. Objective

The objective of this task was to capture live network traffic using *Wireshark* and identify at least three types of network protocols through packet analysis.

2. Tools Used

- **Wireshark (Version X.XX)** – Used for packet capture and protocol analysis
- **Operating System**: Windows 10 (64-bit)

3. Capture Method

- **Network Interface Used**: Wi-Fi (Active adapter for Internet connectivity)
- **Traffic Generation**:
  - Attempted ping google.com with minimal ICMP activity captured
  - Repeated with ping firefox.com which successfully produced ICMP traffic
  - Accessed websites via browser to generate TCP, DNS, and QUIC packets
- **Capture Duration**: 1 minute
- **File Saved As**: task5-network_traffic_capture_and_analysis.pcap

4. Protocols Identified

Protocol	Description	Use Case Example
TCP	Reliable, connection-based protocol in transport layer	Used by most applications that require data delivery accuracy
DNS	Converts domain names into IP addresses	Accessed **firefox.com**, triggering DNS resolution
QUIC	Secure transport protocol over UDP	Seen in encrypted communication from web browsing

Note: QUIC is typically used by modern sites such as YouTube and Google for encrypted communication.

5. Packet Insights

- **TCP**:
  - The first packet captured was a TCP ACK Keep-Alive, indicating a previously established session.
  - No SYN packets were observed, likely due to the timing of the capture.

Protocol	Length	Info
TCP	54	443 → 59039 [FIN, ACK] Seq=25 Ack=2 Win=501 Len=0
TLSv1.2	78	Ignored Unknown Record
TCP	54	59039 → 443 [FIN, ACK] Seq=26 Ack=25 Win=513 Len=0
TCP	54	59039 → 443 [ACK] Seq=27 Ack=26 Win=513 Len=0
TLSv1.2	78	Application Data
TLSv1.2	78	Ignored Unknown Record
TCP	54	443 → 59042 [FIN, ACK] Seq=25 Ack=2 Win=504 Len=0
TCP	54	59042 → 443 [FIN, ACK] Seq=26 Ack=25 Win=510 Len=0
TCP	54	59042 → 443 [ACK] Seq=27 Ack=26 Win=510 Len=0
TCP	66	443 → 59048 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1440 SACK_PERM WS=128
TCP	54	59048 → 443 [ACK] Seq=1 Ack=1 Win=132352 Len=0
TLSv1.2	922	Client Hello (SNI=d.clarity.ms)
TCP	54	443 → 59038 [RST] Seq=25 Win=0 Len=0
TCP	54	443 → 59038 [RST] Seq=25 Win=0 Len=0
TCP	54	443 → 59038 [RST] Seq=26 Win=0 Len=0
TCP	54	443 → 59040 [RST] Seq=26 Win=0 Len=0
TCP	54	443 → 59040 [RST] Seq=26 Win=0 Len=0

- **DNS**:
  - Standard query and response packets followed the initial TCP packet, used for domain resolution (firefox.com).

DNS	97	Standard query 0x8bc2 AAAA fls-na.amazon.com
DNS	298	Standard query response 0x812c A fls-na.amazon.com CNAME gateway.prod.us-east-1.forester.a2z.com CNAME endpoint.prod.u
DNS	247	Standard query response 0x8bc2 AAAA fls-na.amazon.com CNAME gateway.prod.us-east-1.forester.a2z.com CNAME endpoint.pro
DNS	120	Standard query 0xca5c A endpoint.prod.us-east-1.forester.a2z.com
DNS	248	Standard query response 0xca5c A endpoint.prod.us-east-1.forester.a2z.com A 3.226.160.210 A 54.163.20.191 A 18.204.133
DNS	120	Standard query 0xa08e AAAA endpoint.prod.us-east-1.forester.a2z.com
DNS	120	Standard query response 0xa08e AAAA endpoint.prod.us-east-1.forester.a2z.com
DNS	92	Standard query 0xa8ba A d.clarity.ms
DNS	92	Standard query 0xb20d AAAA d.clarity.ms
DNS	155	Standard query response 0xb20d AAAA d.clarity.ms CNAME vmss-clarity-ingest-eus.eastus.cloudapp.azure.com
DNS	171	Standard query response 0xa8ba A d.clarity.ms CNAME vmss-clarity-ingest-eus.eastus.cloudapp.azure.com A 57.151.77.234
DNS	92	Standard query 0x9e22 AAAA d.clarity.ms
DNS	129	Standard query 0xd80a A vmss-clarity-ingest-eus.eastus.cloudapp.azure.com
DNS	155	Standard query response 0x9e22 AAAA d.clarity.ms CNAME vmss-clarity-ingest-eus.eastus.cloudapp.azure.com
DNS	145	Standard query response 0xd80a A vmss-clarity-ingest-eus.eastus.cloudapp.azure.com A 57.151.77.234
DNS	129	Standard query 0xc032 AAAA vmss-clarity-ingest-eus.eastus.cloudapp.azure.com
DNS	129	Standard query response 0xc032 AAAA vmss-clarity-ingest-eus.eastus.cloudapp.azure.com

- **QUIC**:
  - Identified through traffic over UDP port 443, linked to encrypted web traffic.

Protocol	Length	Info
QUIC	1262	Handshake, DCID=638e03, SCID=03142003f2e0fb9b794dfa73cfddfe5c01cf7ed8
QUIC	1262	Handshake, DCID=638e03, SCID=03142003f2e0fb9b794dfa73cfddfe5c01cf7ed8
QUIC	603	Handshake, DCID=638e03, SCID=03142003f2e0fb9b794dfa73cfddfe5c01cf7ed8
QUIC	117	Handshake, DCID=03142003f2e0fb9b794dfa73cfddfe5c01cf7ed8, SCID=638e03
QUIC	150	Handshake, DCID=03142003f2e0fb9b794dfa73cfddfe5c01cf7ed8, SCID=638e03
QUIC	136	Protected Payload (KP0), DCID=03142003f2e0fb9b794dfa73cfddfe5c01cf7ed8
QUIC	1096	Protected Payload (KP0), DCID=03142003f2e0fb9b794dfa73cfddfe5c01cf7ed8
QUIC	1514	Protected Payload (KP0), DCID=638e03
QUIC	169	Protected Payload (KP0), DCID=638e03
QUIC	105	Protected Payload (KP0), DCID=03142003f2e0fb9b794dfa73cfddfe5c01cf7ed8
QUIC	105	Protected Payload (KP0), DCID=638e03
QUIC	105	Protected Payload (KP0), DCID=03142003f2e0fb9b794dfa73cfddfe5c01cf7ed8
QUIC	105	Protected Payload (KP0), DCID=638e03
QUIC	526	Protected Payload (KP0), DCID=638e03
QUIC	105	Protected Payload (KP0), DCID=638e03
QUIC	105	Protected Payload (KP0), DCID=03142003f2e0fb9b794dfa73cfddfe5c01cf7ed8
QUIC	105	Protected Payload (KP0), DCID=638e03

- **ICMP**:
  - Traffic generated by pinging `firefox.com`

```
C:\Users\Dell>ping firefox.com

Pinging firefox.com [35.190.14.201] with 32 bytes of data:
Reply from 35.190.14.201: bytes=32 time=29ms TTL=112
Reply from 35.190.14.201: bytes=32 time=27ms TTL=112
Reply from 35.190.14.201: bytes=32 time=19ms TTL=112
Reply from 35.190.14.201: bytes=32 time=21ms TTL=112

Ping statistics for 35.190.14.201:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 19ms, Maximum = 29ms, Average = 24ms

C:\Users\Dell>
```

  - Echo Request/Reply packets captured with public IP `104.16.xx.xx` as destination

Protocol	Length	Info
ICMPv6	86	Neighbor Solicitation for 2405:201:6828:c89d::c0a8:1d01 from 1c:bf:ce:09:a9:a0
ICMPv6	78	Neighbor Advertisement 2405:201:6828:c89d::c0a8:1d01 (rtr, sol)

6. Observations

- The capture began with a TCP ACK packet, confirming that recording started during an ongoing session.
- DNS packets followed shortly after, showing domain resolution activity initiated during web browsing.
- QUIC traffic was identified after DNS resolution, indicating the browser established secure connections over UDP.
- Protocol filters tcp, dns, and quic were applied to analyze the relevant traffic streams in Wireshark.
- Screenshots of filtered views for each protocol were captured to support analysis.

7. Conclusion

The task successfully demonstrated practical application of Wireshark for capturing live network traffic and analyzing multiple protocols. DNS, TCP, and QUIC were identified, filtered, and reviewed, fulfilling the objective of hands-on protocol analysis using real-time data.