Wireshark – Network Traffic Analysis Report 1.

Objective:

The purpose of this activity was to capture and analyze real network traffic that happened in my network using Wireshark to understand the behavior of DNS, TCP, HTTP protocols. The analysis helped me to build foundational skills in packet inspection and network forensics for SOC operations.

Environmental Setup:

Tool: Wireshark

Operating System: Windows

Network Type: Wi-Fi (active internet connection)

Capture duration: ~3 minutes.

Websites visited: https://en.wikipedia.org/wiki/Python , www. Youtube.com.

File saved as : capture1.pcap.png

Procedure:

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

2.Started packet capture using Wireshark's fin icon.

3.Visited YouTube and Wikipedia in a web browser.

4. Allowed the capture run for approximately 2-3 minutes.

5.Stopped the capturing using the red square icon and saved the pcap.

6.Reopend the pcap file to analyze specific protocols using filters.

Analysis and observations.

1. DNS analysis.

Filter used: dns

Observations: The capture contained several DNS queries and responses.

Ex: - the system queried the domain Wikipedia.org and received the Ip address 103.102.166.240.

This shows the name resolution process, before establishing the network connection to the destination.

2. TCP analysis.

Filter used: tcp

Observation: TCP three way handshakes were observed between host and the multiple servers.

Sequence of packets: SYN -> SYN/ACK -> ACK confirms successful connection establishment.

And also got PSH, ACK which means send data immediately.

3.HTTPS/HTTP analysis.

Filter used: http and https

Observation: When applying the http and https filters, no result appeared. This is because in moderns' websites like Wikipedia use encrypted HTTPS connections. Wireshark does not label encrypted traffic as HTTP and HTTPS: instead such traffic appears as TLS, TLSv1.2 or TLSv1.3

4. TLS analysis.

Filter used: tls

Observation: For YouTube, the captured packets showed TLS handshakes, encrypted traffic and QUIC traffic.

The handshake packets included "Client Hello" and "Server hello", used to negotiate encryption parameters between client and server. The actual data that transmitted is encrypted, which is typical for secure modern web services. All requests and responds are encrypted, so only the TLS handshake and packet sizes, timing and server address can be seen. Encrypted data packets can be observing under TLSv1.2 and TLSv1.3. in Wireshark.

Additionally, QUIC traffic was observed, which is a modern transport protocol that runs over UDP.This protocol combines the features of TCP, TLS and HTTP/3 to provide faster and reliable connections, while still keeping the data encrypted. This shows why some web traffics may not appear under traditional HTTP/HTTPS filters.

Statistics summary.

Total packets captured: 1440.

Top protocols: DNS, TCP, TLS, QUIC

Conversations: between local IP and external IPs like Google DNS, Wikipedia, YouTube, Chatgpt.

Conclusion of the exercise.

The Wireshark capture highlighted how modern network traffic operates. DNS queries resolved domain names to IPs, TCP handshake ensured reliable connection between client and the server, and the PSH, ACK packets indicate the immediate data transfer. Most web traffic used HTTPS/TLS, with encrypted data visible only as TLS/TLSv1.2/TLSv1.3 packets and QUIC traffic over UDP explained why some traffics did not appear under HTTP/HTTPS filters.