**Capture and Analyze Network Traffic Using Wireshark**

1. Install Wireshark with Npcap support.
2. Selected active interface Wi-Fi and started capture.
3. Generated traffic by:

Browsing to https://example.com

Running ping google.com

1. Continued capture for 60 seconds.
2. Applied display filters (dns, tcp, http) to inspect specific traffic types.
3. Stopped capture and exported to .pcap format.

**Packet Analysis**

**DNS Query**

* **Packet No:** 17
* **Source IP:** 192.168.1.5 → **Destination IP:** 8.8.8.8
* **Protocol:** DNS
* **Info:** Standard query A example.com
* **Transaction ID:** 0x3f72
* **Response Packet No:** 18 (resolved to 93.184.216.34)

**TCP 3-Way Handshake**

**Session:** Client 192.168.1.5 → Server 93.184.216.34 (port 80)

* Packet 21: SYN from client to server
* Packet 22: SYN-ACK from server to client
* Packet 23: ACK from client to server — connection established.

**HTTP GET Request**

* **Packet No:** 24
* **Method:** GET / HTTP/1.1
* **Host:** example.com
* **User-Agent:** Mozilla/5.0 (Windows NT 10.0; Win64; x64)
* **Response (Packet 25):** HTTP/1.1 200 OK, Content-Type: text/html

**Statistics**

**Protocol Hierarchy (Top 5)**:

1. TCP — 165 packets (53%)
2. DNS — 25 packets (8%)
3. HTTP — 10 packets (3%)

**Top Endpoints (IPv4)**:

* 192.168.1.5 — Local host
* 8.8.8.8 — Google Public DNS
* 93.184.216.34 — Example.com server
* 142.250.183.206 — Google web server

**Conclusion**

* The capture contained normal, expected network activity for a system performing web browsing and pings.
* DNS traffic was directed to Google Public DNS (8.8.8.8).
* Most application-layer traffic used TLS encryption, indicating secure HTTPS communication.
* The few HTTP packets were from visiting http://example.com (non-sensitive test site).
* No suspicious traffic or anomalies (e.g., repeated failed handshakes, malformed packets, excessive ARP requests) were detected.