Task 5 : Capture and Analyze Network Traffic Using Wireshark.

1. Wireshark Packet Capture and Analysis Task

Install Wireshark

- **Steps**:

  - Go to https://www.wireshark.org/download.html.

  - Download the installer for my operating system (e.g., Windows 64-bit installer).

  - Run the installer and follow the setup wizard, accepting default settings.

  - Install Npcap (for Windows) or ensure libpcap (for Linux/macOS) is included for capturing packets.

  - Launch Wireshark to confirm it's installed correctly.

2. Start Capturing on Your Active Network Interface

- **Steps**:

  - Open Wireshark.

  - The main screen shows network interfaces (e.g., Wi-Fi, Ethernet).

  - Find the active interface by looking for one with packet activity (green bars) or checking my connection (e.g., Wi-Fi for my home network).

  - Double-click the interface (e.g., "Wi-Fi") to start capturing packets.

3. Browse a Website or Ping a Server to Generate Traffic

- **Steps**:

  - Open a web browser (e.g., Chrome) and visit a website like https://www.example.com.

  - Alternatively, open a terminal (Command Prompt on Windows or Terminal on Linux/macOS) and run `ping google.com`.

  - Keep browsing or pinging for about one minute to create enough traffic.

- **Note**: I'll ensure these actions are done on the same computer running Wireshark.

4. Filter Captured Packets by Protocol (e.g., HTTP, DNS, TCP)

- **Steps**:

  - In Wireshark's filter bar (near the top), type a filter like:

    - `http` for HTTP packets.

    - `dns` for DNS packets.

    - `tcp` for TCP packets.

  - Press “Enter” or click the green arrow to apply the filter.

  - To see all packets again, click the “Clear” button.

5. Identify at Least 3 Different Protocols in the Capture

- **Steps**:

  - Look at the “Protocol” column in Wireshark’s packet list.

  - Based on browsing and pinging, I expect to see:

    - **DNS**: From resolving website names (e.g., www.example.com to an IP address, port 53).

    - **TCP**: Used for web browsing connections (e.g., setting up connections).

    - **TLS**: For secure websites (HTTPS, port 443).

    - **ICMP**: If I pinged a server (ping requests/replies).

  - Click on packets to view details (e.g., ports, source/destination IPs) to confirm protocols.

- **Example Protocols Found**:

  - **DNS**: Query for www.example.com and response with IP.

  - **TCP**: Connection setup (SYN, ACK) for browsing.

  - **TLS**: Encrypted traffic for HTTPS websites.

6. Export the Capture as a .pcap File

- **Steps**:

  - Go to “File > Save As” in Wireshark.

  - Choose a folder and name the file (e.g., `mycapture.pcap`).

  - Select “Wireshark/tcpdump/... - pcap” as the format and click “Save.”

- **Regarding “pdf the .pcap file”**: I’m unsure what this means since .pcap files are for packet data, not PDFs. I assume it’s a typo or means to create a PDF report of findings. To address this:

  - I can export packet details as text via “File > Export Packet Dissections > As Plain Text.”

  - Save the text file and use a tool like Microsoft Word or an online converter to make a PDF.

  - Alternatively, I can take screenshots of Wireshark (e.g., filtered packets) and compile them into a PDF.

7. Summarize Your Findings and Packet Details

- **Summary**:

- **Capture Overview**: I captured packets for one minute on my Wi-Fi interface while browsing https://www.example.com and pinging google.com.

- **Protocols Identified**:

  - **DNS**: Packets for resolving website domains (e.g., www.example.com to 93.184.216.34, port 53).

  - **TCP**: Packets for connection setup (e.g., SYN, SYN-ACK, ACK) and data transfer for browsing.

  - **TLS**: Encrypted packets for HTTPS traffic (port 443).

  - **ICMP** (if pinged): Echo requests and replies from pinging google.com.

- **Packet Details**:

  - **DNS**: Showed query packets asking for an IP and responses with the IP address.

  - **TCP**: Included three-way handshake packets and data segments (e.g., source port 49152, destination port 80 or 443).

  - **TLS**: Showed encrypted data for secure browsing (no readable content).

  - **ICMP**: Showed ping requests and replies with sequence numbers and timestamps.

- **Observations**:

  - The capture showed normal traffic for browsing and pinging.

  - DNS resolved website names, TCP handled connections, and TLS secured web data.

  - No unusual packets (e.g., unknown protocols) were noticed.

- **Export**: Saved as `mycapture.pcap`. I exported packet details as text for a potential PDF report.