Task 5 – Capture and Analyze Network Traffic Using Wireshark

Objective

To capture and analyze live network packets using Wireshark and identify key protocols such as TCP, DNS, and mDNS used during real-time browsing sessions.

Tools Used

- Wireshark
- Operating System: Kali Linux
- Utilities: Chrome browser, Command Prompt (for ping)

Steps Performed

1. Installed Wireshark

Downloaded and installed the latest version from https://www.wireshark.org.

2. Started Network Capture

- Opened Wireshark and selected my active Wi-Fi interface.
- Clicked the "Start Capturing" button to begin real-time traffic analysis.

3. Generated Network Traffic

While capturing:

- Browsed websites: YouTube, Google, GitHub
- Used the terminal to run: ping google.com

4. Stopped Capture

Stopped the capture after ~1 minute using the red square stop button.

5. Filtered Protocols

Applied filters:

- tcp – for TCP sessions
- dns – for DNS queries/responses
- mdns – for multicast DNS used in local discovery

6. Exported Captures

Created protocol-specific packet files:

- task5_capture tcp.pcapng
- task5_capture dns.pcapng

- task5_capture MDNS youtube.pcapng
- task5_capture DNS youtube.pcapng

Protocols Identified

Protocol	Description	Observation
TCP	Ensures reliable, ordered delivery of data between client and server.	TCP packets to 172.217.x.x during Google and YouTube access.
DNS	Resolves domain names like google.com to IP addresses.	Queries such as Standard query A google.com
mDNS	Used to resolve hostnames on local networks without a DNS server.	Queries like _services. _ dns-sd. _ udp. local captured during YouTube session

Key Learnings

- Captured real traffic to/from websites like **Google** and **YouTube**
- Understood how to analyze packet-level details like source/destination IPs, ports, flags
- Learned how to apply protocol filters in Wireshark (TCP, DNS, HTTP, MDNS)
- Differentiated between external DNS resolution and local mDNS service discovery

Outcome

Successfully captured and analyzed multiple networking protocols using Wireshark—exported and documented real-time internet traffic, including visits to Google and YouTube.