# Task 5: Capturing and Analyzing Network Traffic Using Wireshark

## 🎯 Goal

The purpose of this task is to monitor live network activity and study different types of network protocols by capturing and analyzing packets using Wireshark.

## 🧰 Tools Used

Wireshark (installed on Kali Linux VM)

## 🪜 Step-by-Step Guide

Note: This task was executed on a Kali Linux Virtual Machine.

### 1️⃣ Open Wireshark

You can start Wireshark using either method:

Terminal command:

wireshark

GUI path:
Applications → Sniffing & Spoofing → Wireshark

### 2️⃣ Choose a Network Interface

Select your currently active interface like eth0, ens33, or wlan0, then click the Start Capturing button (blue shark fin icon).

### 3️⃣ Create Some Network Activity

To generate real-time traffic:

Open Firefox/Chromium and visit a webpage (e.g., http://example.com)

Or run a command in terminal like:

ping google.com

Let the capture run for about 60 seconds.

### 4 Stop Packet Capture

After enough traffic is collected, stop the capture using the red square Stop button in Wireshark.

### 5 Apply Protocol Filters

Use Wireshark's filter bar to isolate specific types of traffic. Example filters:

http
dns
icmp
tcp

### 6 Protocols Identified

During the capture session, the following protocols were detected:

DNS: Resolves domain names to IP addresses
HTTP: Handles unencrypted web traffic
ICMP: Used for ping requests and echo replies

### 7 Save the Capture File

Go to:
File → Save As
Save the packet data in .pcap format for future analysis or documentation.

## Traffic Analysis Summary

### Protocols Observed

- HTTP – Transfers web content (unencrypted).
- DNS – Translates domains to IPs.
- TCP – Ensures reliable data transfer.
- UDP – Lightweight protocol used by services like DNS.
- ICMPv6 – IPv6 support for ping and neighbor discovery.

### Protocol Insights

- HTTP: Requests/responses from browsing non-HTTPS sites.

**Paavan Shastri**

- DNS: Shows domain lookup queries (A, AAAA records).
- TCP: Used in secure web sessions or for reliable communication.
- UDP: Common in DNS or streaming-related protocols.
- ICMPv6: Found in IPv6 networks for ping or router communication.

## Behavioral Observations

- Web traffic triggered DNS lookups.
- TCP and UDP indicate mixed reliability needs.
- ICMPv6 shows IPv6 services were active during testing.
- Packet count reflects both system and user-originated traffic.