

# Basic Packet Inspection Using Wireshark

## Objective

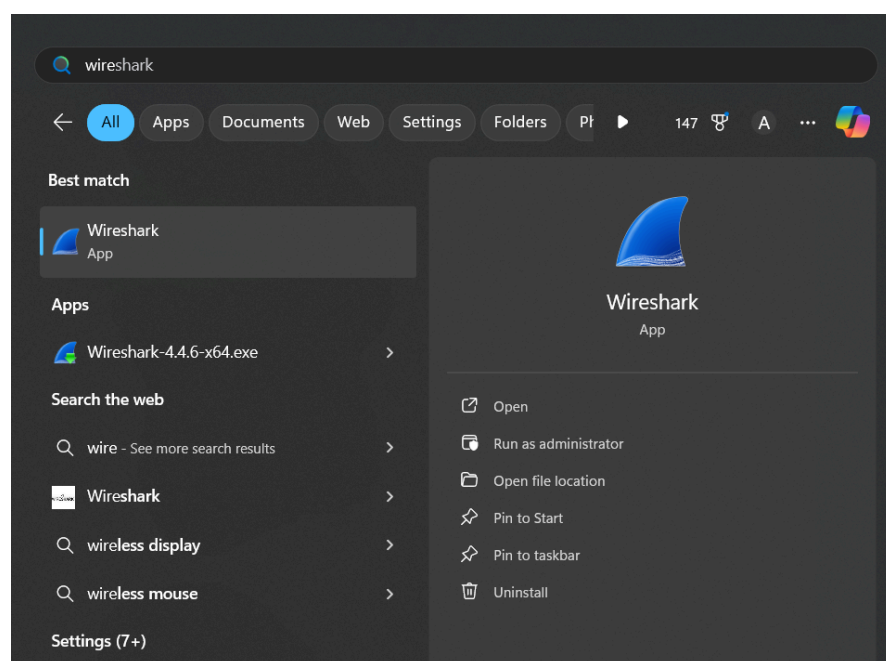
To capture and analyze network traffic using **Wireshark** to understand how common protocols (HTTP, DNS, ARP, TCP, UDP, FTP, IP, SMTP) work and how data is transmitted and received.

## 1. Tools Used

- **Wireshark** (a free and open-source packet analyzer)
- A working internet connection
- Optional: a web browser or command-line tools to generate protocol traffic (e.g., **ping**, **ftp**, **telnet**, email clients)

## 2. Capturing Packets with Wireshark

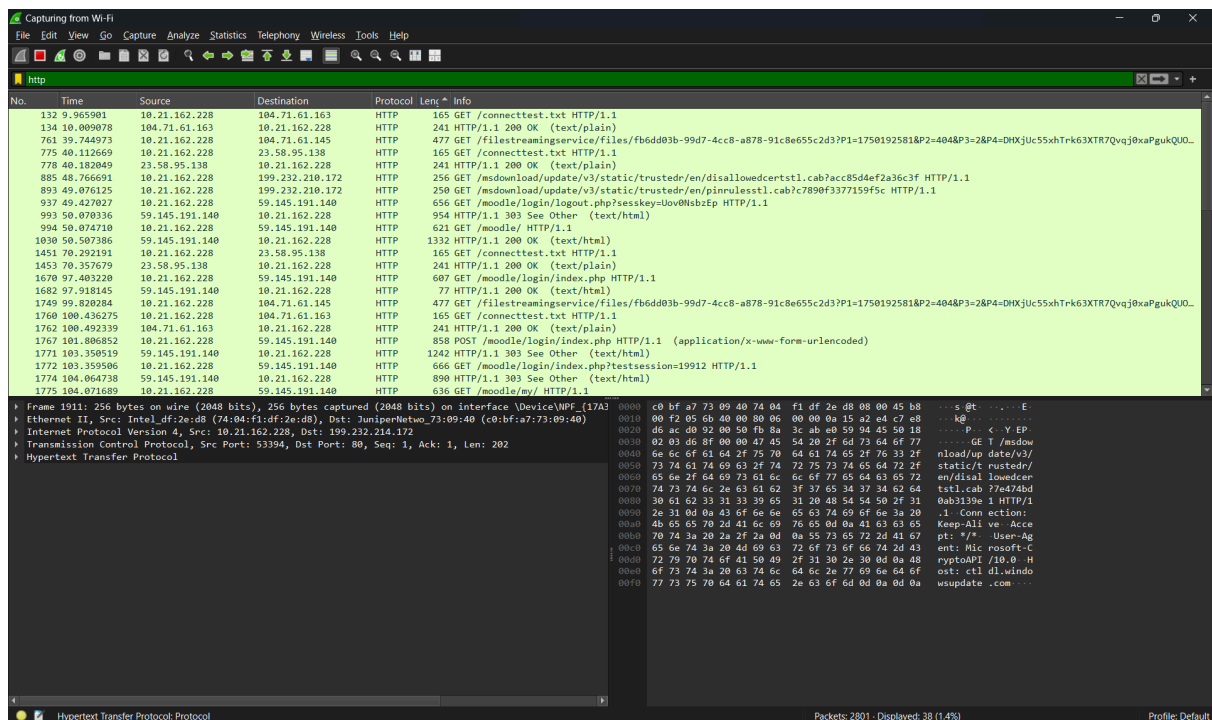
1. Open Wireshark.
2. Select the active network interface (e.g., Wi-Fi or Ethernet).
3. Click **Start Capturing Packets**.
4. Perform actions like visiting websites, sending emails, or using FTP to generate protocol traffic.
5. Click **Stop** after collecting sufficient data.



## 3. Protocol Analysis

### A. HTTP (Hypertext Transfer Protocol)

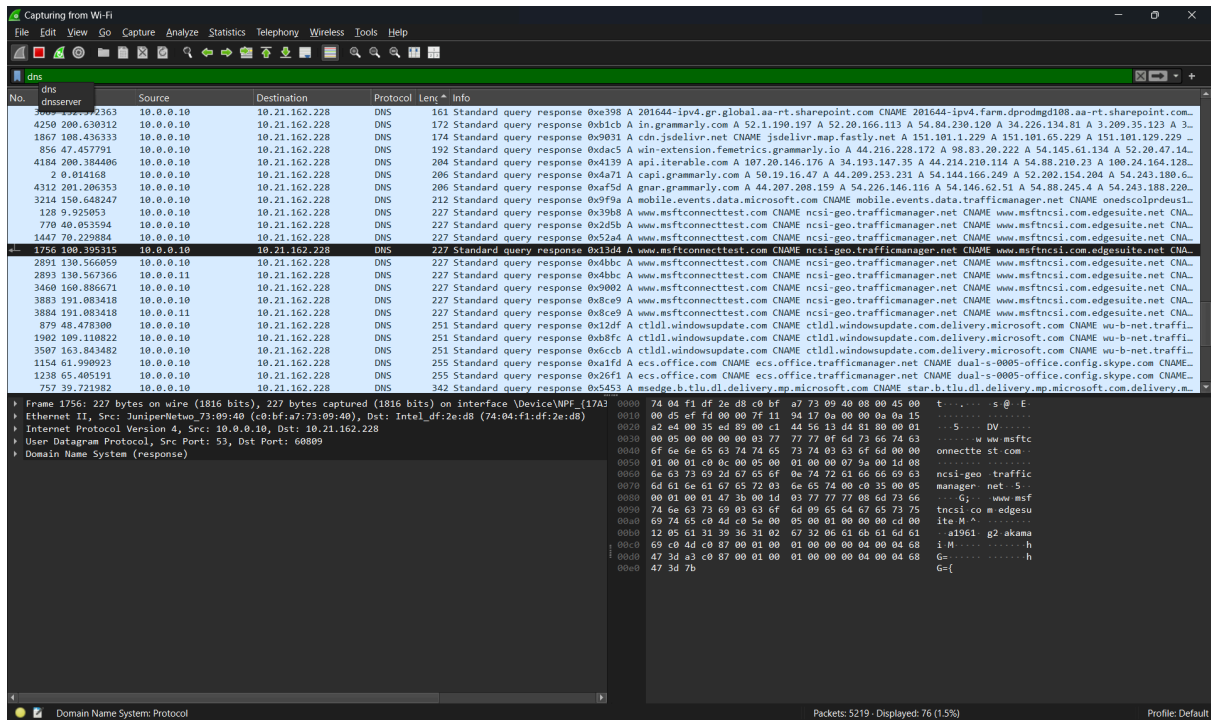
- **Port:** 80
- **Use:** Web browsing (fetching websites)
- **How to identify in Wireshark:** Filter by **http**
- **Look for:**
  - GET/POST requests
  - Response headers (e.g., **200 OK**)
  - Hostname and URI



### B. DNS (Domain Name System)

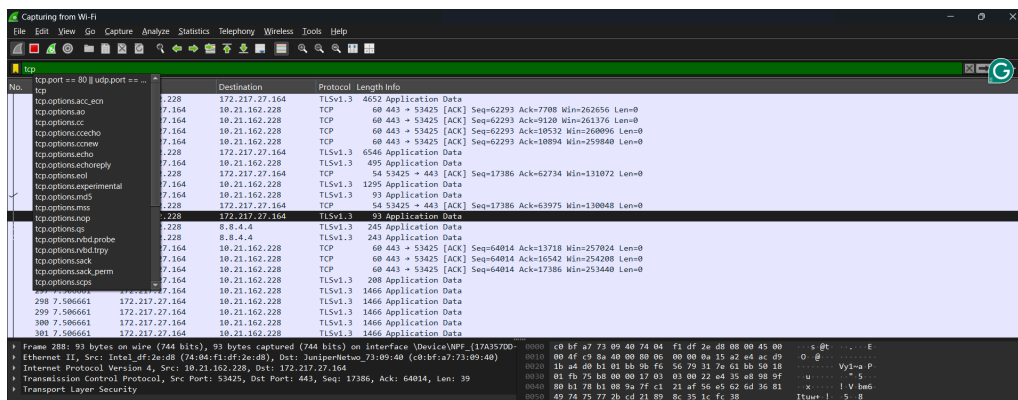
- **Port:** 53 (UDP or TCP)
- **Use:** Translates domain names to IP addresses

- Filter: **dns**
- Look for:
  - DNS queries (**Standard query**)
  - Responses with IP addresses



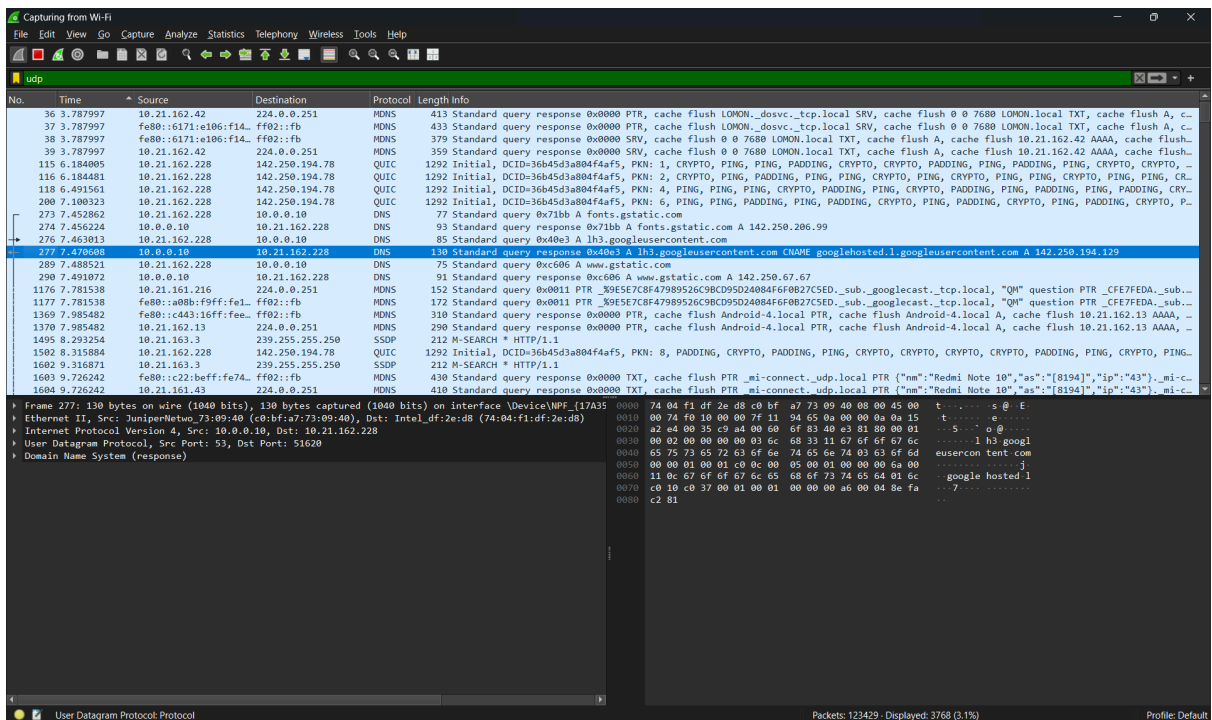
## D. TCP (Transmission Control Protocol)

- Use: Reliable, connection-oriented communication
- Filter: **tcp**
- Look for:
  - 3-way handshake (**SYN**, **SYN-ACK**, **ACK**)
  - Sequence and acknowledgment numbers



## E. UDP (User Datagram Protocol)

- **Use:** Faster, connectionless communication (e.g., video streaming)
- **Filter:** **udp**
- **Look for:**
  - Lightweight packets without handshakes
  - Often used with DNS, VoIP, streaming services



## F. FTP (File Transfer Protocol)

- **Port:** 21 (control), 20 (data)
- **Use:** File transfers between client and server
- **Filter:** **ftp**
- **Look for:**
  - Login commands (**USER**, **PASS**)

- File upload/download commands (**STOR**, **RETR**)
- 

### G. IP (Internet Protocol)

- **Use:** Logical addressing for routing packets
- **Filter:** **ip**
- **Look for:**
  - Source and destination IP addresses
  - IP version, header length, TTL, checksum

### H. SMTP (Simple Mail Transfer Protocol)

- **Port:** 25 (sometimes 587 or 465)
- **Use:** Sending emails from client to server
- **Filter:** **smtp**
- **Look for:**
  - Commands like **HELO**, **MAIL FROM**, **RCPT TO**, **DATA**
  - Email content in plain text

---

## 4. Example Workflow

1. Visit **http://example.com** in your browser.
2. Use **nslookup example.com** to trigger DNS request.
3. Send a test email via command-line or client.
4. Use FTP client to upload/download a file.
5. Analyze each protocol using Wireshark filters.

---

## 5. Observations and Insights

- **DNS** typically precedes HTTP as domains are resolved before browsing.

- **TCP handshakes** establish a reliable session before actual data (HTTP, FTP) is sent.
- **SMTP traffic** is often readable in plain text unless encrypted with TLS.
- **ARP** shows how local MAC addresses are resolved.
- **UDP** lacks acknowledgment, suitable for speed-critical applications.