

DAY 5 - TASK 5

Objective: Capture live network packets and identify basic protocols and traffic types.

Tools: Wireshark (free).

Deliverables: A packet capture (.pcap) file and a short report of protocols identified

Wireshark Protocol Analysis Report

Date of Capture: 11th Aug, 2025

Capture File: [packet_capture.pcap](#)

Duration: 1 minute

Interface Used: eth0

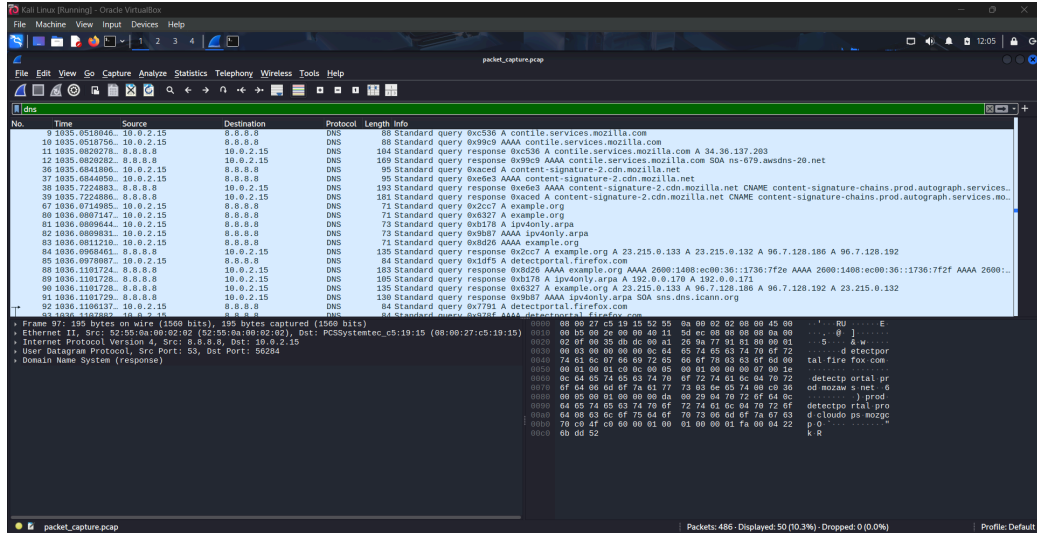
1. Overview

A network packet capture was performed using Wireshark to observe real-time network activity on the Kali Linux VM. The objective was to identify and analyze different network protocols involved in typical browsing activity.

2. Protocols Identified

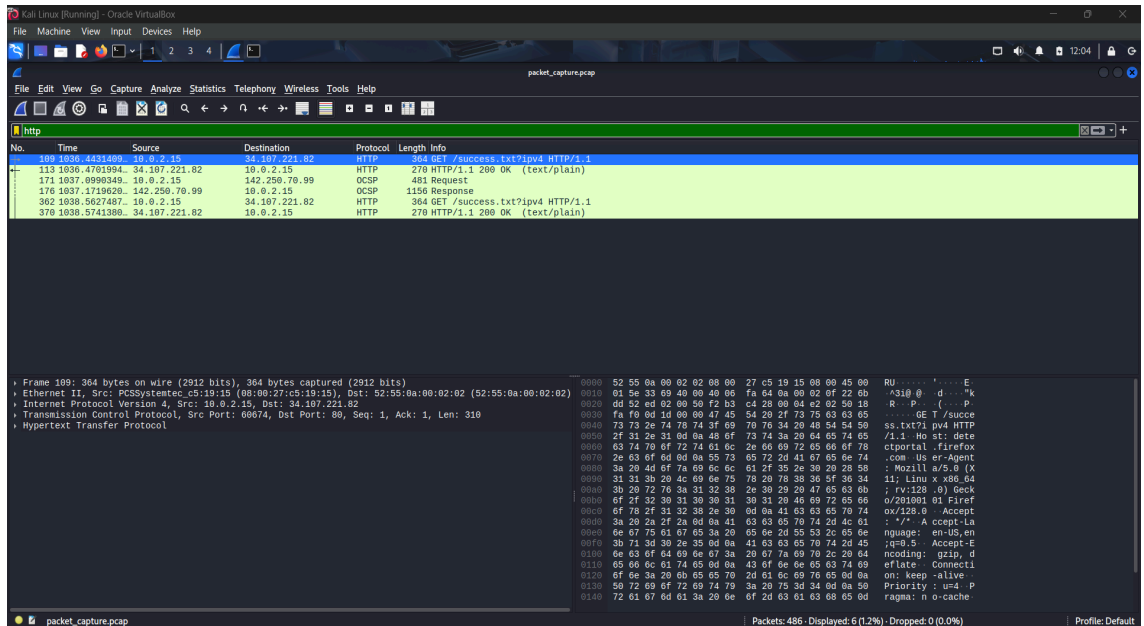
A. DNS (Domain Name System)

- **Purpose:** DNS translates human-readable domain names (e.g., [google.com](#)) into IP addresses for routing over the Internet.
- **Observation in Capture:**
 - Multiple **DNS Query (Standard Query A)** packets were sent from the local machine to the DNS server.
 - **Response packets** contained resolved IP addresses.
- **Port Used:** UDP 53 (sometimes TCP 53 for large responses).



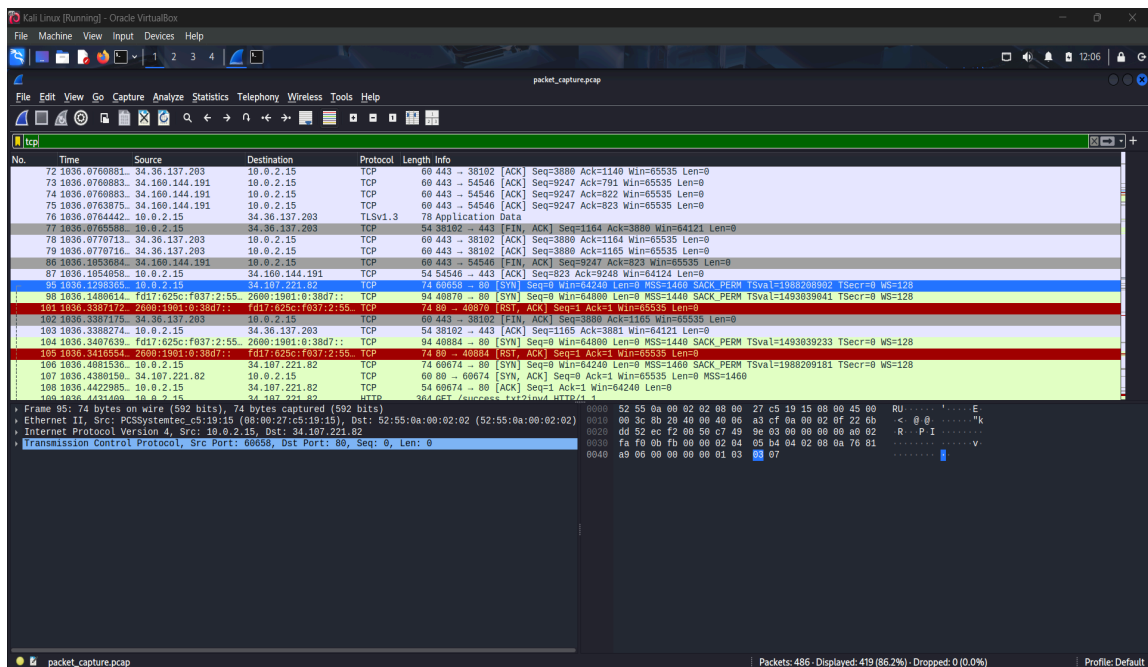
B. HTTP (HyperText Transfer Protocol)

- **Purpose:** HTTP is used for transferring web pages and other web resources.
- **Observation in Capture:**
 - GET requests from the client to the server for HTML content.
 - Response headers visible in the payload.
- **Port Used:** TCP 80



C. TCP (Transmission Control Protocol)

- **Purpose:** TCP provides reliable, ordered, and error-checked delivery of data. It underlies many application protocols like HTTP, HTTPS, and FTP.
- **Observation in Capture:**
 - Multiple TCP 3-way handshakes (**SYN**, **SYN-ACK**, **ACK**).
 - Data segments carrying HTTP and DNS traffic.
 - Connection termination packets (**FIN**, **ACK**).
- **Ports Used:** Varies depending on the application protocol (e.g., 80 for HTTP, 53 for DNS if using TCP).



No.	Time	Source	Destination	Protocol	Length	Info
72	10.36.0760081	34.36.137.203	10.0.2.15	TCP	60	443 → 38102 [ACK] Seq=3880 Ack=1140 Win=65535 Len=0
73	10.36.0760883	34.160.144.191	10.0.2.15	TCP	60	443 → 54540 [ACK] Seq=9247 Ack=791 Win=65535 Len=0
74	10.36.0760883	34.160.144.191	10.0.2.15	TCP	60	443 → 54540 [ACK] Seq=9247 Ack=822 Win=65535 Len=0
75	10.36.0763875	34.160.144.191	10.0.2.15	TCP	60	443 → 54540 [ACK] Seq=9247 Ack=823 Win=65535 Len=0
76	10.36.0764442	10.0.2.15	34.36.137.203	TLSv1.3	78	Application Data
77	10.36.0765586	10.0.2.15	34.36.137.203	TCP	54	38102 → 443 [FIN, ACK] Seq=1164 Ack=3880 Win=64121 Len=0
78	10.36.0770713	34.36.137.203	10.0.2.15	TCP	60	443 → 38102 [ACK] Seq=3880 Ack=1164 Win=65535 Len=0
79	10.36.0770716	34.36.137.203	10.0.2.15	TCP	60	443 → 38102 [ACK] Seq=3880 Ack=1165 Win=65535 Len=0
86	10.36.1053684	34.160.144.191	10.0.2.15	TCP	60	443 → 54540 [FIN, ACK] Seq=9247 Ack=823 Win=65535 Len=0
87	10.36.1054058	10.0.2.15	34.160.144.191	TCP	54	54540 → 443 [ACK] Seq=923 Ack=9248 Win=64124 Len=0
93	10.36.1282355	10.0.2.15	34.107.221.82	TCP	74	60674 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1988708902 TSecr=0 WS=128
98	10.36.1480614	fd17:625c:f037:2:55::	2600:1901:0:38d7::	TCP	94	40870 → 80 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 SACK_PERM TSval=1493039041 TSecr=0 WS=128
101	10.36.1387172	2600:1901:0:38d7::	fd17:625c:f037:2:55::	TCP	74	80 → 40870 [RST, ACK] Seq=1 Ack=1 Win=65535 Len=0
102	10.36.1387175	34.36.137.203	10.0.2.15	TCP	60	443 → 38102 [FIN, ACK] Seq=3880 Ack=1164 Win=65535 Len=0
103	10.36.1388274	10.0.2.15	34.36.137.203	TCP	54	38102 → 443 [ACK] Seq=1165 Ack=3881 Win=64121 Len=0
104	10.36.1407639	fd17:625c:f037:2:55::	2600:1901:0:38d7::	TCP	94	40884 → 80 [SYN] Seq=0 Win=64800 Len=0 MSS=1440 SACK_PERM TSval=1493039233 TSecr=0 WS=128
105	10.36.1410587	2600:1901:0:38d7::	fd17:625c:f037:2:55::	TCP	74	80 → 40884 [RST, ACK] Seq=1 Ack=1 Win=65535 Len=0
106	10.36.1401536	10.0.2.15	34.107.221.82	TCP	74	60674 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=1988709181 TSecr=0 WS=128
107	10.36.14300150	34.107.221.82	10.0.2.15	TCP	60	80 → 60674 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
108	10.36.1442985	10.0.2.15	34.107.221.82	TCP	54	60674 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
109	10.36.1443400	10.0.2.15	34.107.221.82	HTTP	364	GET /facebook-javascript-1.721044-HTTP-1-...

Frame 95: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0
Ethernet II, Src: PCSystemec-c5:19:15 (08:00:27:c5:19:15), Dst: 52:55:0a:09:02:02 (52:55:0a:09:02:02)
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 34.107.221.82
Transmission Control Protocol, Src Port: 60658, Dst Port: 80, Seq: 0, Len: 0

3. Additional Protocols (if observed)

- **ARP (Address Resolution Protocol):** Resolves IP addresses to MAC addresses within the local network.
- **ICMP (Internet Control Message Protocol):** Used for ping requests and network diagnostics.

