Task 3: Network Packet Capture and Analysis

Goal

Capture and analyze network packets to identify protocols and understand traffic behavior.

Tools Used

- **Wireshark**
- **Kali Linux**
- **Apache Web Server (DVWA – localhost)**

Environment

- Operating System: Kali Linux
- Target: Localhost (`127.0.0.1`)
- Network Interface: Loopback (`lo`)

Procedure

1. TCP Traffic Capture

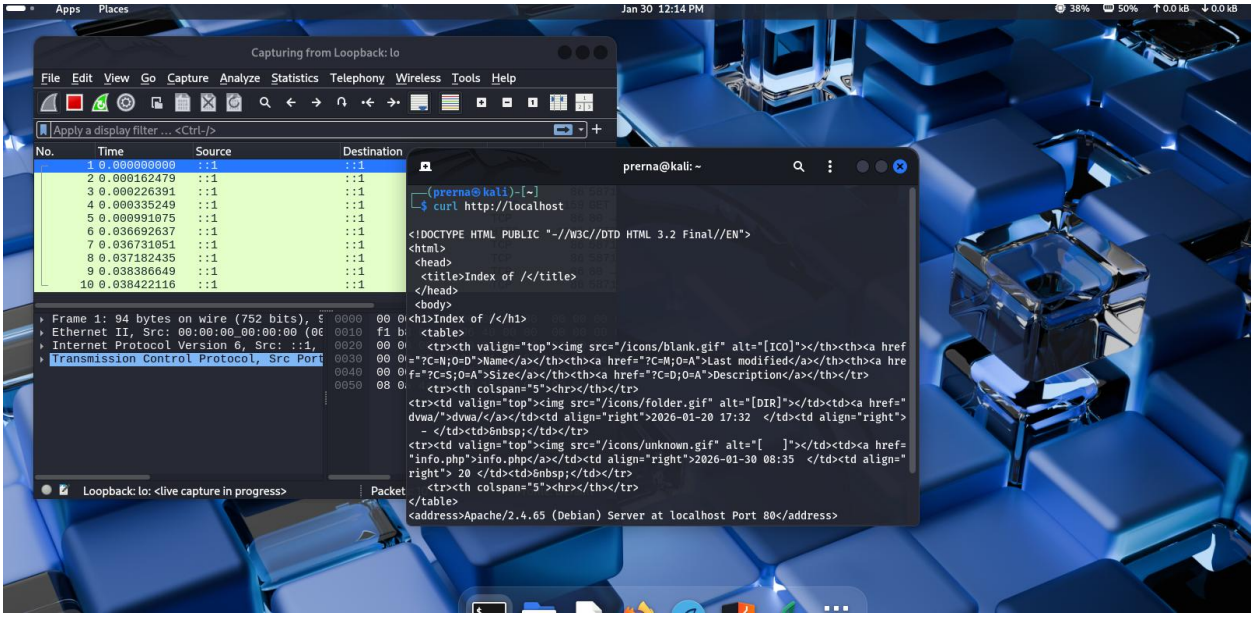
Wireshark was started on the loopback (`lo`) interface to capture localhost traffic.
An HTTP request was generated using the following command:

```
curl http://127.0.0.1/dvwa
```

A display filter `tcp` was applied in Wireshark to observe TCP packets.
The TCP three-way handshake was identified, consisting of:

- SYN
- SYN-ACK
- ACK

These packets confirmed successful establishment of a TCP connection between the client and the local web server.



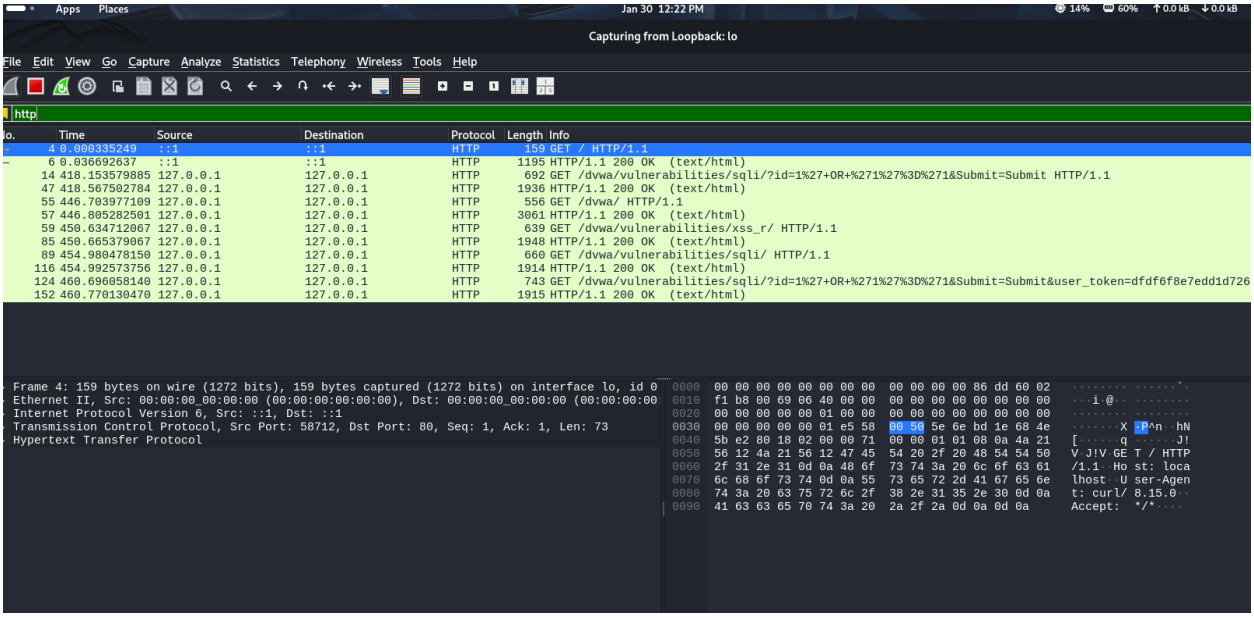
2. HTTP Traffic Capture

With the capture still running, HTTP traffic was generated by accessing the DVWA web application on localhost.

A display filter `http` was applied in Wireshark.
The captured packets included:

- HTTP GET requests for DVWA pages
- HTTP responses such as **200 OK**

The packet details showed HTTP headers, request methods, and server responses, confirming successful HTTP communication.



Analysis

- **TCP Traffic:**
  The TCP handshake packets demonstrated reliable connection establishment before data transfer.
- **HTTP Traffic:**
  HTTP GET requests and responses confirmed web communication over the established TCP connection.
- Since the application was hosted locally, the loopback (`lo`) interface was used, which is appropriate for analyzing localhost traffic.

Conclusion

Wireshark successfully captured and analyzed TCP and HTTP traffic generated by a local web application. The analysis confirmed proper TCP connection establishment and HTTP request-response communication, fulfilling the task requirements.