NETWORK PACKET CAPTURE & TCP ANALYSIS REPORT

Analyst: Precious Anyanwu
Date: *24/01/2026*
Case ID: NET-ANALYSIS-001

---

1. Case Summary

This lab exercise demonstrates how data packets are transmitted and captured within a local test environment. The objective was to observe real network communication, understand the TCP three-way handshake process, and analyze how a client and server establish and maintain a connection.

A local HTTP server was configured, and traffic between the client and server was monitored using packet capture tools.

---

2. Lab Environment

Component	Description
OS	Kali Linux
Capture Tool	tcpdump
Server Tool	Python HTTP Server
Client Tool	Netcat (nc)
Network Interface	Loopback Interface (lo)

---

3. Tools Used

- **tcpdump** – Network packet capture and analysis

- **Python HTTP Server** – To simulate a local web server
- **Netcat (nc)** – To simulate client connection requests

---

4. Packet Capture Setup

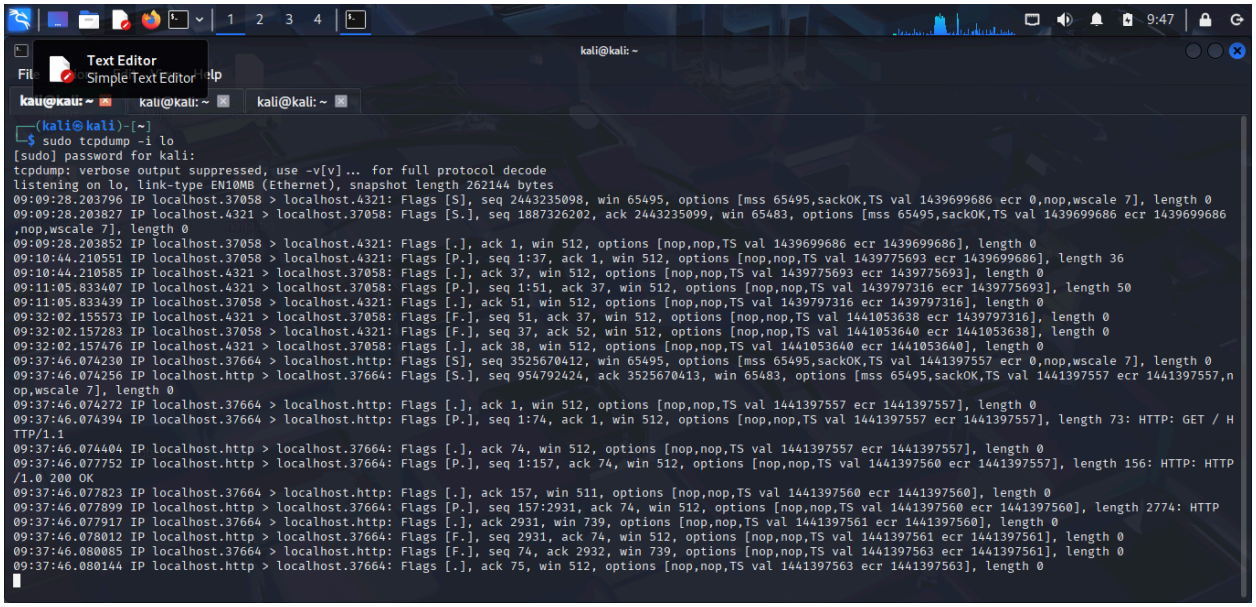
To begin monitoring traffic, the available network interfaces were listed:

tcpdump -D

The loopback interface was selected for capture:

sudo tcpdump -i lo

This command captures packets flowing through the local system (127.0.0.1), allowing observation of client-server communication within the machine.



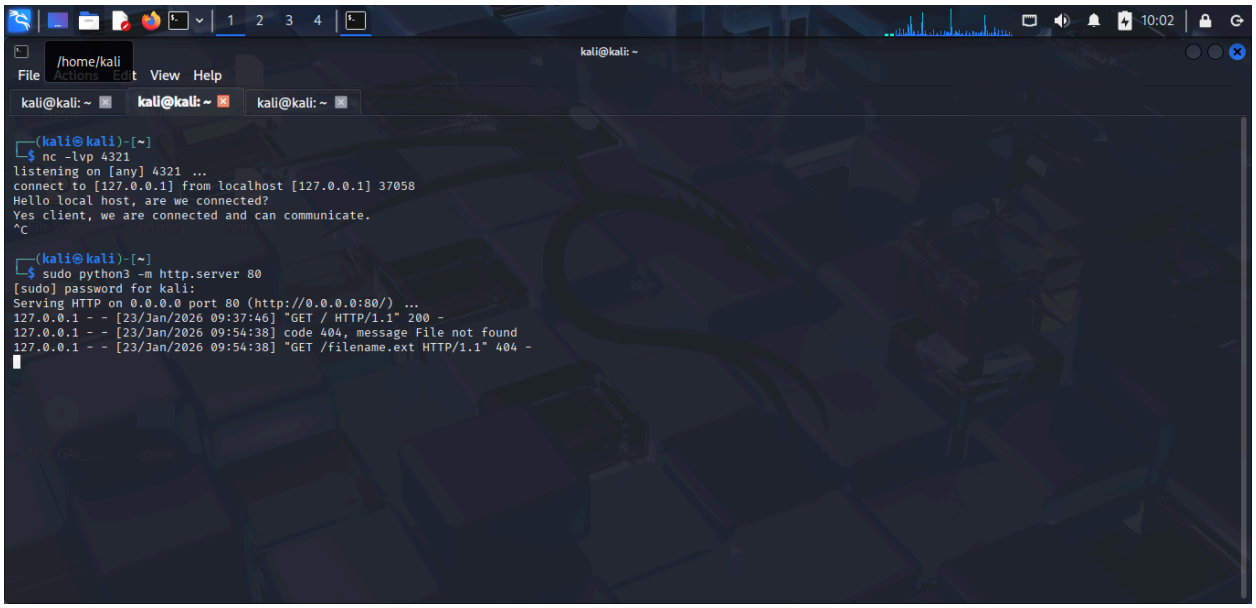
---

5. Server Configuration

A local HTTP server was created on port 80:

sudo python3 -m http.server 80

This allowed the system to act as a web server, waiting for incoming client requests.



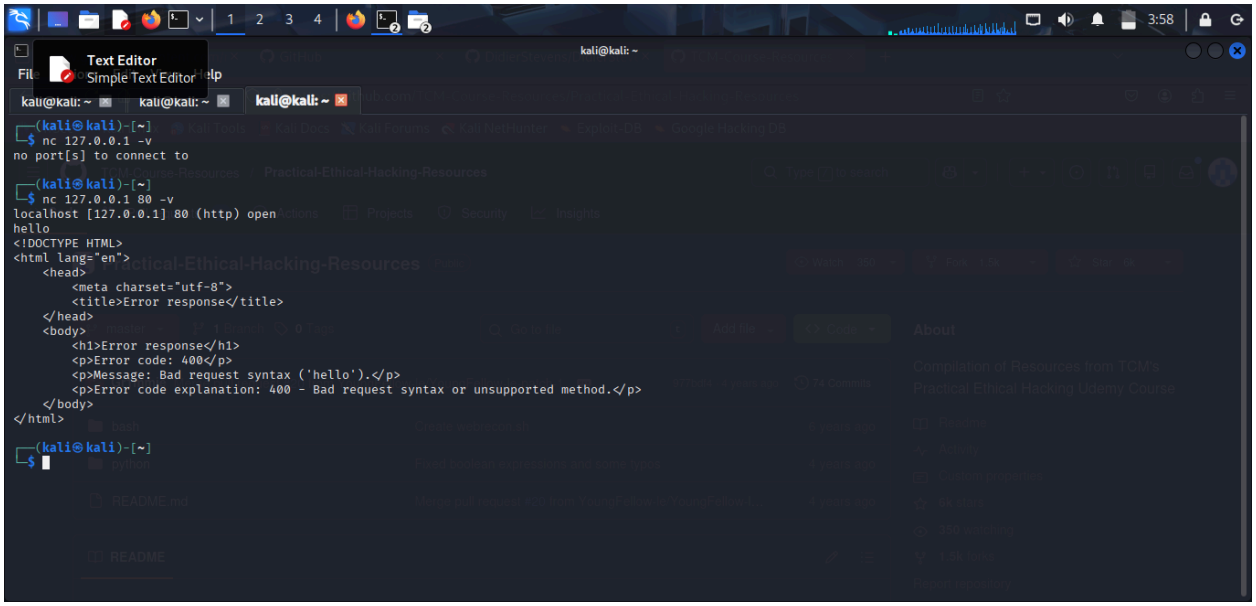
---

6. Client Connection

A client was created using Netcat to initiate communication with the server:

nc 127.0.0.1 80 -v

This command established a connection between the client and the local HTTP server.



---

7. TCP Three-Way Handshake Observation

The packet capture displayed the TCP three-way handshake, which establishes reliable communication between systems:

1. **SYN** – The client sends a synchronization packet with an initial sequence number.
2. **SYN-ACK** – The server responds, acknowledging the client's sequence number and sending its own.
3. **ACK** – The client acknowledges the server's sequence number.

After this exchange, the TCP session is established, and data transfer begins.

This process confirms TCP's connection-oriented nature and reliability.

---

8. Traffic Analysis Test

To generate additional traffic, a request was made for a non-existent file:

curl http://127.0.0.1/filename.ext -O filename.ext

This action produced observable HTTP request and response packets in tcpdump, demonstrating how application-layer requests appear in packet captures.

(Insert Screenshot: tcpdump showing request/response)

---

9. Key Observations

- Successful capture of TCP handshake packets
- Sequence and acknowledgment numbers incremented correctly
- HTTP request traffic was visible in packet capture
- Demonstrated how local traffic flows through the loopback interface

---

10. Conclusion

This lab provided practical exposure to real network packet flow and TCP communication behavior. Observing the three-way handshake reinforced understanding of TCP as a reliable, connection-oriented protocol that ensures ordered and controlled data delivery.

Hands-on packet analysis is a foundational skill in network security and SOC operations, where analysts monitor traffic to detect anomalies, malicious communication, or policy violations.