S.Balamurugan
241901014

EXPT: 6 IMPLEMENTATION OF PACKET SNIFFING USING RAW SOCKETS IN PYTHON

Aim:

Create a basic Python script that uses a raw socket to capture network packets and print details such as source/destination IP, protocol type, TCP/UDP ports, and a short hexadecimal view of the payload.

Algorithm:

1. Open a raw **AF_PACKET** or raw socket to capture all incoming packets.
2. Continuously receive packets in a loop.
3. Read and decode the Ethernet header; if the packet contains IPv4, extract the IP header.
4. If the packet uses TCP or UDP, extract port numbers and print a short summary along with a hex dump of the data.
5. Keep capturing until the user stops the program using **Ctrl + C**.

Code:

```
from scapy.all import sniff, IP, TCP, UDP, Raw

def packet_callback(packet):
    print("\n--- Packet Captured ---")

    if IP in packet:
        print(f"Source IP: {packet[IP].src}")
        print(f"Destination IP: {packet[IP].dst}")
        print(f"Protocol: {packet[IP].proto}")

        if TCP in packet:
            print(f"Source Port: {packet[TCP].sport}")
            print(f"Destination Port: {packet[TCP].dport}")

        if UDP in packet:
            print(f"Source Port: {packet[UDP].sport}")
            print(f"Destination Port: {packet[UDP].dport}")

        if Raw in packet:
            print("Payload (Hex):", packet[Raw].load[:32].hex().upper())
        else:
            print("No payload data.")

print("Starting Packet Sniffer... Press Ctrl+C to stop.")
```

```
sniff(prn=packet_callback, store=False)


from scapy.all import sniff, IP, TCP, UDP, Raw

def packet_callback(packet):
    print("\n--- Packet Captured ---")

    if IP in packet:
        print(f"Source IP: {packet[IP].src}")
        print(f"Destination IP: {packet[IP].dst}")
        print(f"Protocol: {packet[IP].proto}")

        if TCP in packet:
            print(f"Source Port: {packet[TCP].sport}")
            print(f"Destination Port: {packet[TCP].dport}")

        if UDP in packet:
            print(f"Source Port: {packet[UDP].sport}")
            print(f"Destination Port: {packet[UDP].dport}")

        if Raw in packet:
            print("Payload (Hex):", packet[Raw].load[:32].hex().upper())
        else:
            print("No payload data.")

print("Starting Packet Sniffer... Press Ctrl+C to stop.")
sniff(prn=packet_callback, store=False)
```

Output:

```
PS H:\bala> python .\PACKET_SNIFFING.py
Starting Packet Sniffer... Press Ctrl+C to stop.

--- Packet Captured ---
Source IP: 192.168.29.190
Destination IP: 18.138.163.31
Protocol: 17
Source Port: 55602
Destination Port: 53
Payload (Hex): 00000000

--- Packet Captured ---
Source IP: 18.138.163.31
Destination IP: 192.168.29.190
Protocol: 17
Source Port: 53
Destination Port: 55602
Payload (Hex): A23F983B98

--- Packet Captured ---

--- Packet Captured ---

--- Packet Captured ---
Source IP: 192.168.29.1
Destination IP: 224.0.0.1
Protocol: 2
Payload (Hex): 1164E5870000000009140000
```

Result:

The raw socket packet-sniffing script ran successfully.
It detected live network packets and printed information such as source and destination IPs, protocol types, port numbers, and a short hexadecimal preview of the packet content.