EX: 14

Data: 29/10/24

Raw Socket to implement Packet Sniffing

Aim:

TO write code using RAW sockets to implement Packet sniffing

Algorithm:

→ start sniffing: Capture Packets on specified network interface

→ check Packet type. for each Packet check if it IP Packet

→ Identify Protocol: Determine the protocol

→ Display

→ repeat

First initial scapy

→ Pip install scapy

Code:

Packet_Sniffer.py

```
from scapy.all import sniff
from scapy.layers.inet import IP, TCP, UDP, ICMP
def Packet_callback(Packet):
    if IP in PACKET[IP]
    Protocol = IP_Layer.Proto
    SRC_IP = IP_layer.SRC
    dst_IP = IP_Layer.dst
```

```
    Protocol_name = " "
    if Protocol == 1:
        Protocol_name = "ICMP"

    elif protocol == 6:
        Protocol_name = "TCP"
    elif Protocol == 17:
        Protocol_name = "UDP"

    else:
        Protocol_name = "UnKnown Protocol"
    Print(f"Protocol : {Protocol}")
    Print(f"Source IP: {src_ip}")
    Print(f"destination IP: {dst_IP}")
    Print("_" * 50)

def main():
    sniff(iface='wifi', prn=Packet_callback, filter="ip",
                                                store=0)

if __name__ == "__main__":
    main()
```

Output:

Protocol: TCP
Source IP: 51.132.193.105
Destination IP: 192.168.34.193
_ _ _ _ _ _ _ _ _ _ _ _ _
Protocol: TCP
Source IP: 192.168.34.193
Destination IP: 51.132.193.65

Result:

Thus implementation of Packet Sniffing is done using Raw sockets