Ex:12
7/10/25

# Implementing Packet sniffing using RAW Socket

## AIM:

To implment packet sniffing using RAW Socket

## Algorithm:

- * check for root privilages & open a new socket bound to choosen network interface
- * receive raw frames from select (recvfrom) in a loop
- * Ethernet header from select in a loop
- * Parse Ethernet header to extract source MAC destination MAC, Ethertype.
- * if Ethertype == IPv4, parse the IPv4 header to get version, TTL, protocol, source IP destination
- * print summary
- * Repeat until stopped, then close socket & Exit cleanly.

## Code:

```
def packet_callback (Packet):
        if IP in packet:
            ip_layer = packet [IP]
            Protocol = ip_layer.proto
```

```
src-ip = ip_layer. src
dst-ip = ip_ layer. dst

Protocal_ name = " "
if protocal == 1
        Protocal_name = "ICMP"
elif Protocal == 6:
        Protocal_name = "TCP
elif protocal == 17:
        Protocol _ name= "UDP"
else:
        Protocal_name = "unknown protocal"
Print (f"Protocal: {Protocal_name}"}
Print (f" Source IP: {src-ip}")
Print (f"Destination IP: {dst_ip}")
Print ("-" * 50)

sniff ( face - "wifi" , Prn = Packet-callback,
filter = "ip" , store = 0)
```

Input:

Pinging a server (ping)

Output :

```
Protocal : TCP
source IP: 192.168.1,5
Destination IP : 172.217.15.78
- - - - - - - - - -
Protocol : ICMP
Source IP: 192.168-1.5
Destination: IP: 8.8.8.8
- - - - - - - - - - -
Protocal : UDP
Source IP: 192.168.1.5
Destination IP: 224.0.0.251
```

9

Result:

Packet sniffing using Raw socket is Implemented & Executed.