EX NO: 14
DATE: 25/10/24

Practical-14

AIM:-
write a code using RAW sockets to implement Packet sniffing.

ALGORITHM:
1] Initialize a raw socket to capture all incoming packets on a network interface
2] Bind it to network interface.
3] Continuously receive packets and store them in buffer
4) Extract and display packet details, such as IP headers and Protocol.
5] Repeat until stopped, allowing real-time Packet monitoring

SOURCE CODE:-
```
from scapy.all import sniff
from scapy.layers.inet import IP, TCP, UDP, ICMP

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
        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_name}")
print(f"Source IP: {src_ip}")
print(f"Destination IP: {dst_ip}")
print("_" * 50)

def main():
    sniff(iface='Wi-Fi', prn=Packet_callback, filter="IP", store=0)
if __name__ == "__main__":
    main()
```

OUTPUT:-

```
Protocol: TCP
Source IP: 20.247.184.142
Destination: 172.20.10.2

Protocol: TCP
Source IP: 20.247.184.142
Destination: 172.20.10.2
```

RESULT:-

Thus, RAW sockets to implement Packet sniffing is implemented and executed successfully.