

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 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 + ("Protocol : {protocol_name}")

print + ("Source IP: {src_ip}")

print + ("Destination IP: {dst_ip}")

print("-" * 50)

def main():

sniff(interface = 'wif1', prn=Packet_callback, filter='IP', store=0)

if __name__ == "__main__":

main()

OUTPUT:-

Protocol: TCP

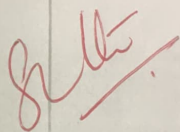
Source IP: 20.247.184.142

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.