PACKET SNIFFING

Ex: 14
Date: 25/10/24

AIM: Write a code using RAW sockets to implement packet sniffing.

ALGORITHM:

Step 1: start script
Step 2: Import modules (scapy functions and classes)
Step 3: Define callback function (packet - callback)
if packet has IP layer, extract protocol and IPs
Identify protocol type using conditions
print protocol, source and destination IP.
Step 4: Define main() to start sniffing on a given interface.
Step 5: Run main() if script is run directly.
Step 6: Terminate on user interrupt.
Step 7: Stop.

CODE:

```
from scapy.all import sniff
from scapy.layers.inet import IP, TCP, UDP, ICMP

def packet_callback(packet):
    if IP in packet[IP]
        ip_layer = packet[IP]
        protocol = ip_layer.proto
        src_ip = ip_layer.src
        dst_ip = ip_layer.dst

# Determine the protocol
    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 packet details
    print(f"Protocol: {protocol_name}")
    print(f"Source IP: {src_ip}")
    print(f"Destination IP: {dst_ip}")
    print("-" * 40)

def main():
    sniff(iface='WiFi', 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 the program is executed and the output is verified