

Implementing of Packet Sniffing Using RAW Sockets

6/10/25
Exp: 14

Aim :- To implement packet sniffing using RAW Sockets

Algorithm :-

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(f"Protocol: {protocol_name}")

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

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

print("-" * 50)

Sniff Capture = filter

Sample Input

Step 1: Open

https://www

Step 2: Run

analyzer

Sample Output

Protocol:

Source IP:

Destination IP:

RES

7

be

Sniff Ciface = 'wi-fi', from-pocket callback,
filter = "ip", nstr = 0)

Sample Input :-

Step 1: Open a web browser and visit

https://www.google.com

Step 2: Run the command 8.8.8.8 ping in
another Terminal

Sample output :-

Protocol : TCP

Source IP : 192.168.1.5

Destination IP : 142.250.183.110

RESULT :- Hence the experiment on the implementation
of Packet sniffing using RAW sockets has
been executed successfully