NAME:THULASI.S EXPERIMENT:6

ROLL:NO:241901118 DATE:01/9/25

IMPLEMENT PACKET SNIFFING USING RAW SOCKETS IN PYTHON

AIM:

To develop a python program that captures and analyzes network packets using raw sockets,along with packet payload data.

INTRODUCTION:

Packet sniffing is a way to watch and capture data that moves across a computer network. It helps to see what information is being sent and received between devices.

ALGORITHM:

1.Create a new socket bound to network interface to capture all packets.

2.Receive packets continuously from network interface.

3.Extract and parse the ethernet header to get source MAC,destination MAC,and protocol type.

4.If protocol indicates IPV4,parse the IP header to extract IP version.

5.If IP protocol is TCP,parse the TCP header.

6.Extract and display any data payload in hexadecimal format.

7.Repeat the process until manually stopped.

CODE:

```
import socket
import struct
import binascii
import textwrap

def main():
    # Get host
    host = socket.gethostbyname(socket.gethostname())
    print('IP: {}'.format(host))

    # Create a raw socket and bind it
    conn = socket.socket(socket.AF_INET, socket.SOCK_RAW,
socket.IPPROTO_IP)
    conn.bind((host, 0))

    # Include IP headers
    conn.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)
    # Enable promiscuous mode
    conn.ioctl(socket.SIO_RCVALL, socket.RCVALL_ON)

    while True:
        # Recive data
        raw_data, addr = conn.recvfrom(65536)

        # Unpack data
        dest_mac, src_mac, eth_proto, data = ethernet_frame(raw_data)

        print('\nEthernet Frame:')
        print("Destination MAC: {}".format(dest_mac))
        print("Source MAC: {}".format(src_mac))
        print("Protocol: {}".format(eth_proto))
def ethernet_frame(data):
    dest_mac, src_mac, proto = struct.unpack('!6s6s2s', data[:14])
    return get_mac_addr(dest_mac), get_mac_addr(src_mac),
get_protocol(proto), data[14:]
def get_mac_addr(bytes_addr):
    bytes_str = map('{:02x}'.format, bytes_addr)
    mac_address = ':'.join(bytes_str).upper()
    return mac_address
```

```
    def get_protocol(bytes_proto):
        bytes_str = map('{:02x}'.format, bytes_proto)
        protocol = ''.join(bytes_str).upper()
        return protocol
    main()
```

OUTPUT:

```
(ctf) →  packetSniffing sudo python sniffing.py
IP: 192.168.1.7

Ethernet Frame:
Destination MAC: 45:00:00:3C:33:DB
Source MAC: 40:00:40:06:2C:60
Protocol: 6466

Ethernet Frame:
Destination MAC: 96:3E:F2:DF:53:F2
Source MAC: F0:ED:B8:04:89:78
Protocol: 86DD

Ethernet Frame:
Destination MAC: 96:3E:F2:DF:53:F2
Source MAC: F0:ED:B8:04:89:78
Protocol: 86DD

Ethernet Frame:
Destination MAC: F0:ED:B8:04:89:78
Source MAC: 96:3E:F2:DF:53:F2
Protocol: 86DD

Ethernet Frame:
Destination MAC: 33:33:00:00:00:02
Source MAC: 96:3E:F2:DF:53:F2
Protocol: 86DD

Ethernet Frame:
Destination MAC: 01:00:5E:00:00:16
Source MAC: 96:3E:F2:DF:53:F2
Protocol: 0800
```

RESULT:

The program has successfully implemented the packet sniffing using raw sockets in python.