NAME: SIVARANJANII.M

DATE: 02.09.25   ROLL NO.:241901109

EXERCISE 6

IMPLEMENT PACKET SNIFFING USING RAW SOCKETS IN PYTHON

PACKET SNIFFING:

Packet sniffing is the process of capturing and analyzing network data packets to monitor traffic, troubleshoot networks, or detect security issues.

AIM:

   To develop a Python program using raw sockets that captures Ethernet frames and displays their Source MAC, Destination MAC, and Protocol information.

ALGORITHM:

1. Get the host IP address.
2. Create a raw socket and bind it to the host.
3. Enable IP headers and promiscuous mode.
4. Continuously receive packets from the network.
5. For each packet
   - Extract destination MAC, source MAC, and protocol.
   - Convert them into human-readable format.
   - Display the results.

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)
```

```python
        # 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))
# Unpack ethernet frame
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:]
# Return formatted MAC address AA:BB:CC:DD:EE:FF
def get_mac_addr(bytes_addr):
    bytes_str = map('{:02x}'.format, bytes_addr)
    mac_address = ':'.join(bytes_str).upper()
    return mac_address
# Return formatted protocol ABCD
def get_protocol(bytes_proto):
    bytes_str = map('{:02x}'.format, bytes_proto)
    protocol = ''.join(bytes_str).upper()
    return protocol
main()
```

OUTPUT:

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>python sniffing.py
IP: 172.20.10.9

Ethernet Frame:
Destination MAC: 45:00:00:28:23:AE
Source MAC: 40:00:80:06:00:00
Protocol: AC14

Ethernet Frame:
Destination MAC: 45:00:01:92:23:AF
Source MAC: 40:00:80:06:00:00
Protocol: AC14

Ethernet Frame:
Destination MAC: 45:00:01:06:23:B0
Source MAC: 40:00:80:06:00:00
Protocol: AC14

Ethernet Frame:
Destination MAC: 45:00:03:7A:23:B1
Source MAC: 40:00:80:06:00:00
Protocol: AC14

Ethernet Frame:
Destination MAC: 45:00:00:28:23:B2
Source MAC: 40:00:80:06:00:00
Protocol: AC14

Ethernet Frame:
Destination MAC: 45:00:00:28:AE:29
Source MAC: 40:00:80:06:00:00
Protocol: AC14
```

RESULT:

The program successfully captures network packets and displays their MAC addresses and protocol details, demonstrating basic packet sniffing functionality.