EX.NO:5

PACKAGE SNIFFING USING RAW SOCKETS

AIM:

To write a python program that captures network packets using raw sockets and display details like source MAC address,destination MAC address,protocol.

WHAT IS PACKET?

A **packet** is a small unit of data transmitted over a network. It contains both the data being sent and control information, like source and destination addresses, to ensure it reaches its destination correctly.

PROCEDURE:

1.Start python and import required modules(socket,struct,binascii).

2.Get the IP address of the host computer.

3.Create a raw socket and bind it to the best.

4.Enable the socket to include headers and turn on promiscuous mode (to capture all packetd).

5.Continuously receive packets from the network.

6.For each packet:

i)Extract the ethernet frame.

ii)Read and display the destination MAC address,source MAC address,and protocol.

7.Keep repeating to capture more packets will the program is stopped.

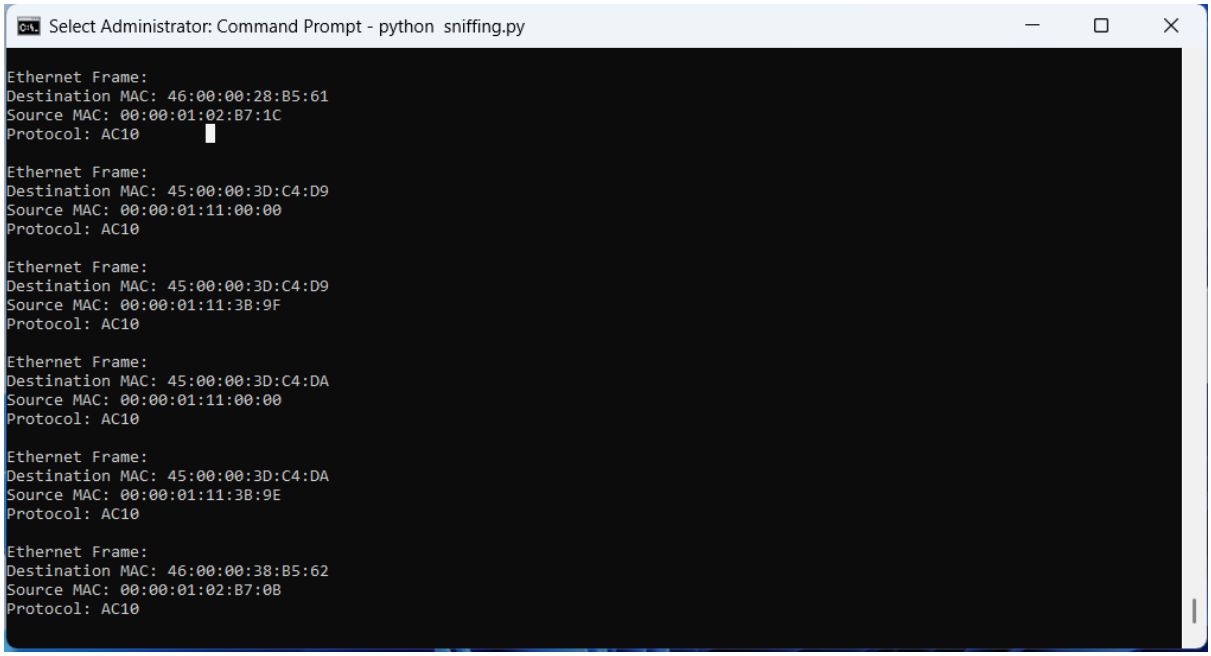
CODE:

```
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))
```

```
# 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:



RESULT:

Thus program shows the source and destination MAC address and protocol of network packets its captures.

NAME :DHARANI K

ROLL NO :241901025

DEPARTMENT:CSE-CYBER SECURITY