Exp:no:4

Roll no:241901023

Name: Dhanalakshmi.C

Department: CSE-Cyber Security

IMPLEMENT PACKET SNIFFING USING RAW SOCKETS IN PYTHON

AIM:

To capture and display network packets received by the host, parse Ethernet/IP headers and show source/destination addresses and protocol information for packet analysis and learning.

PROCEDURE:

1.Get the local IP address of the machine to bind the socket.

2.Create a raw socket to capture all incoming packets at the network interface.

3.Bind the socket to the IP address and set options to include IP headers.

4.Enable promiscuous mode so the network adapter captures all packets, not just those meant for your PC.

5.Receive packets continuously from the network interface.

6.Unpack the Ethernet frame to extract

- Destination MAC address
- Source MAC address
- Protocal type

7.Formal and display these details for each captured packet.

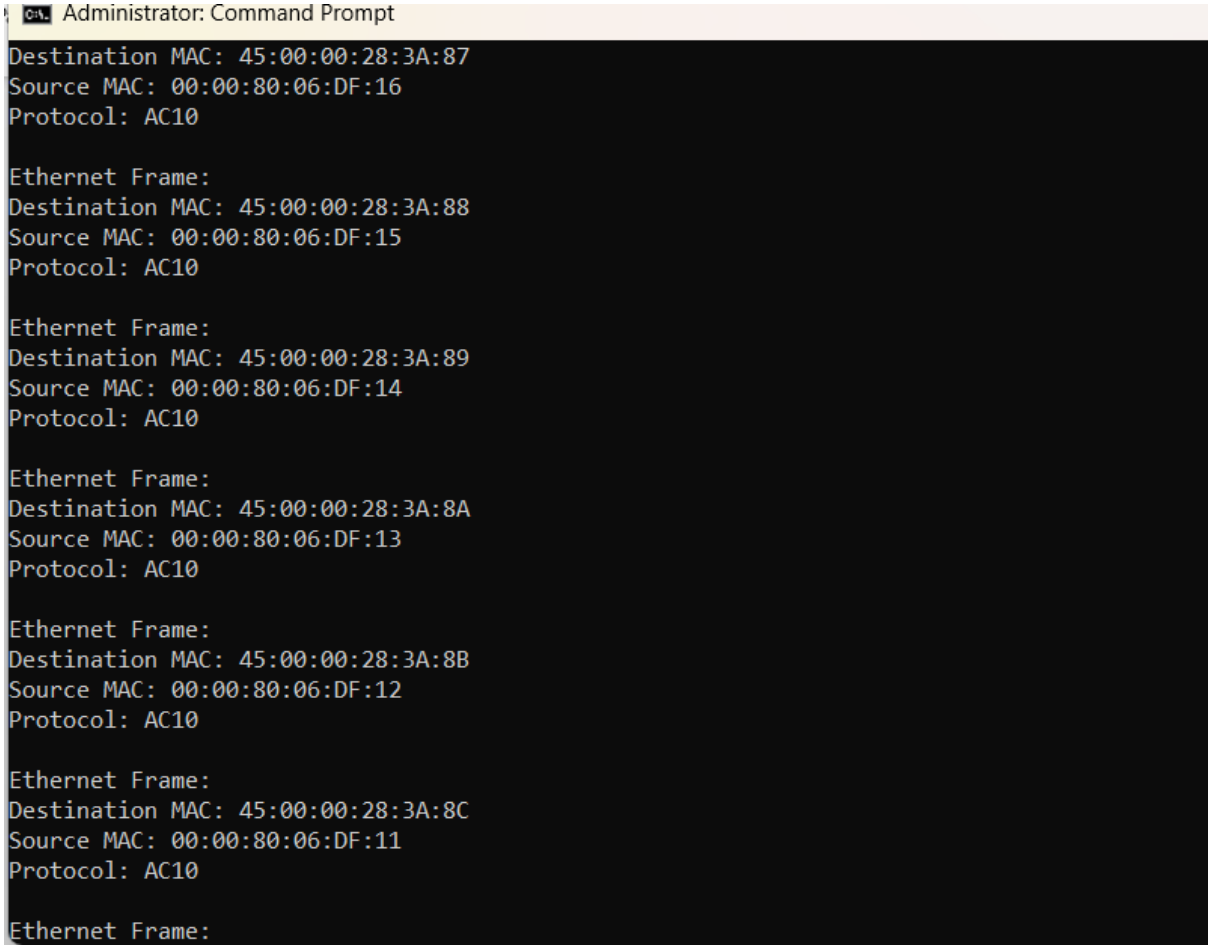
8.Repeat the capture until stopped manually or a certain condition is met.

PROGRAM:

```
import socket
import struct
import binascii
import textwrap
def main():
   host = socket.gethostbyname(socket.gethostname())
   print('IP: {}'.format(host))
   conn = socket.socket(socket.AF_INET, socket.SOCK_RAW, socket.IPPROTO_IP)
   conn.bind((host, 0))
   conn.setsockopt(socket.IPPROTO_IP, socket.IP_HDRINCL, 1)
   conn.ioctl(socket.SIO_RCVALL, socket.RCVALL_ON)
   while True:
      raw_data, addr = conn.recvfrom(65536)
      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:



RESULT:

The program shows the source and destination MAC address and protocol of network packets it captures.