**Task 1: Basic Network Sniffer**

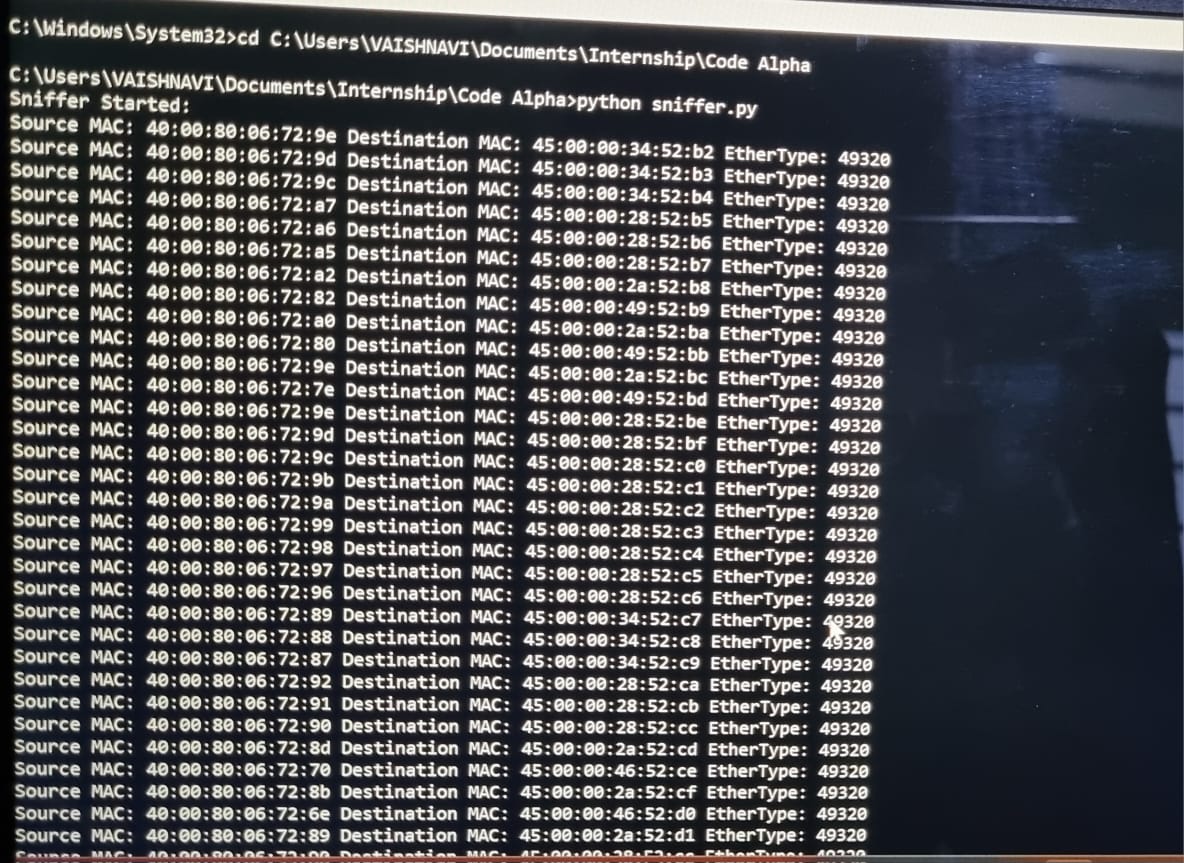
**Problem Statement:**

Build a network sniffer in Python that captures and analyzes network traffic. This project will help you understand how data flows on a network and how network packets are structured.

**Python Code:**

import socketimport structclass IPHeader: def \_\_init\_\_(self, raw\_data): ip\_header = struct.unpack('!BBHHHBBH4s4s', raw\_data) self.protocol = ip\_header[6] self.source\_ip = socket.inet\_ntoa(ip\_header[8]) self.destination\_ip = socket.inet\_ntoa(ip\_header[9])def conn(): # Create a raw socket and bind it to the network interface sniffer = socket.socket(socket.AF\_INET, socket.SOCK\_RAW, socket.IPPROTO\_IP) sniffer.bind(("192.168.90.14", 0)) # Include IP headers sniffer.setsockopt(socket.IPPROTO\_IP, socket.IP\_HDRINCL, 1) # Enable promiscuous mode sniffer.ioctl(socket.SIO\_RCVALL, socket.RCVALL\_ON) return snifferdef main(): sniffer = conn() print("Sniffer Started:") # Sniff packets sniff(sniffer)def sniff(conn): while True: # Receive packet raw\_data, \_ = conn.recvfrom(65536) # Extract Ethernet header (first 14 bytes) eth\_header = raw\_data[:14] # Unpack Ethernet header dest\_mac, src\_mac, eth\_proto = struct.unpack('!6s6sH', eth\_header) # Print MAC addresses and Ethernet protocol print(f"Source MAC: {get\_mac\_address(src\_mac)} Destination MAC: {get\_mac\_address(dest\_mac)} EtherType: {eth\_proto}")def get\_mac\_address(mac): return ":".join("{:02x}".format(b) for b in mac)if \_\_name\_\_ == '\_\_main\_\_': main()

**OUTPUT:**

****