

TASK 1

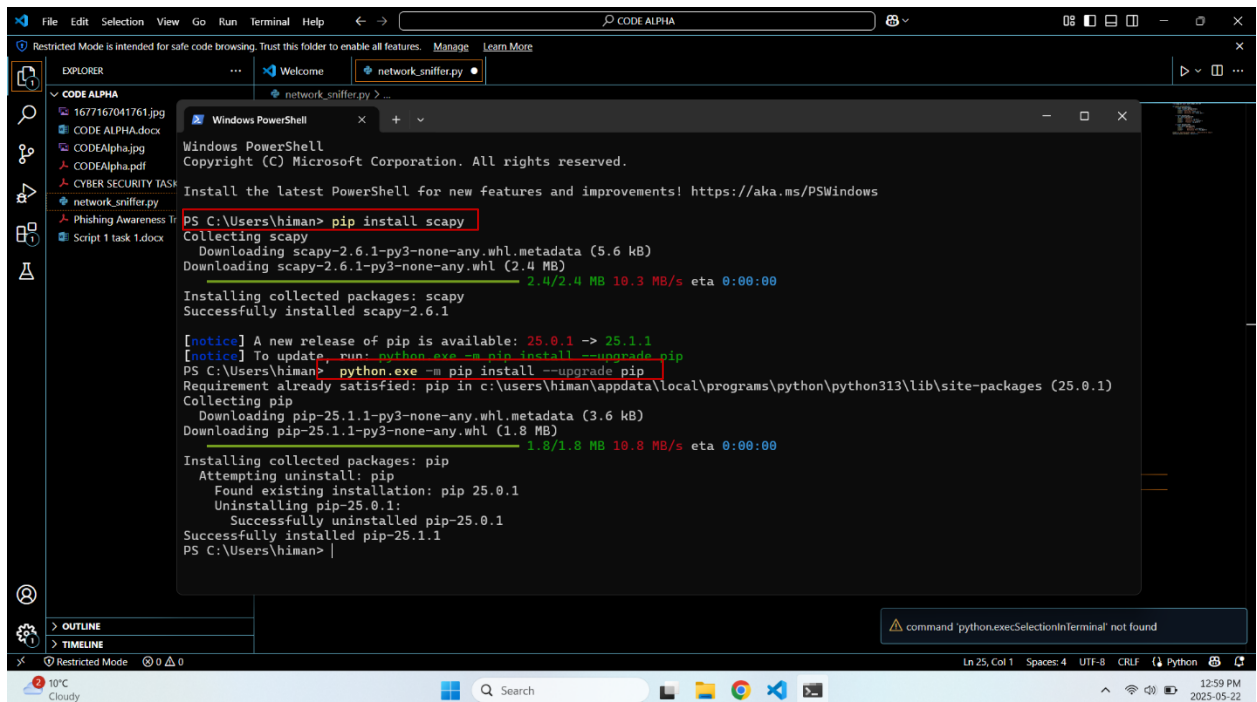
BASIC NETWORK SNIFFER ON WINDOW

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.

➤ Build a Network Sniffer on **Windows** using **VS Code + Scapy**

1. Install Required Software

- **Python 3:** Download and install from <https://www.python.org/downloads/>
- **VS Code:** Download and install from <https://code.visualstudio.com/>
- **Scapy library:**
Open PowerShell or VS Code terminal and run: `pip install scapy`



The screenshot shows the Visual Studio Code (VS Code) interface with a file explorer on the left showing a project named 'CODE ALPHA'. The main editor area displays a file named 'network_sniffer.py'. A Windows PowerShell terminal window is open, showing the command `pip install scapy` being executed. The terminal output shows the successful installation of scapy-2.6.1. Below this, a notice indicates a new release of pip is available (25.0.1 -> 25.1.1). The user then runs `python.exe -m pip install --upgrade pip`, which successfully upgrades pip to version 25.1.1. The terminal output is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

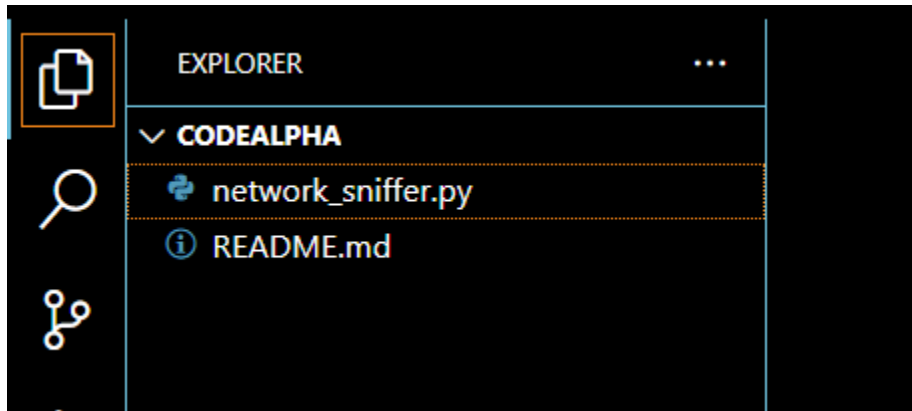
PS C:\Users\himan> pip install scapy
Collecting scapy
  Downloading scapy-2.6.1-py3-none-any.whl.metadata (5.6 kB)
  Downloading scapy-2.6.1-py3-none-any.whl (2.4 MB)
    2.4/2.4 MB 10.3 MB/s eta 0:00:00
Installing collected packages: scapy
Successfully installed scapy-2.6.1

[notice] A new release of pip is available: 25.0.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\himan> python.exe -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\himan\appdata\local\programs\python\python313\lib\site-packages (25.0.1)
Collecting pip
  Downloading pip-25.1.1-py3-none-any.whl.metadata (3.6 kB)
  Downloading pip-25.1.1-py3-none-any.whl (1.8 MB)
    1.8/1.8 MB 10.8 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 25.0.1
    Uninstalling pip-25.0.1:
      Successfully uninstalled pip-25.0.1
  Successfully installed pip-25.1.1
PS C:\Users\himan>
```

2. Write the Sniffer Script

In VS Code:

1. Create a folder (e.g., codealpha_tasks)
2. Inside it, create a new file: `network_sniffer.py`



3. Paste this code below:

```
from scapy.all import sniff, Ether, IP, TCP

def packet_callback(packet):
    if packet.haslayer(Ether):
        ether = packet.getlayer(Ether)
        print(f"\nEthernet Frame:")
        print(f" Source MAC: {ether.src}")
        print(f" Destination MAC: {ether.dst}")

    if packet.haslayer(IP):
        ip = packet.getlayer(IP)
        print(f" IP Packet:")
        print(f" Source IP: {ip.src}")
        print(f" Destination IP: {ip.dst}")
        print(f" Protocol: {ip.proto}")

    if packet.haslayer(TCP):
        tcp = packet.getlayer(TCP)
        print(f" TCP Segment:")
        print(f" Source Port: {tcp.sport}")
        print(f" Destination Port: {tcp.dport}")

print("[*] Starting packet capture... Press Ctrl+C to stop.")
sniff(prn=packet_callback, store=False)
```



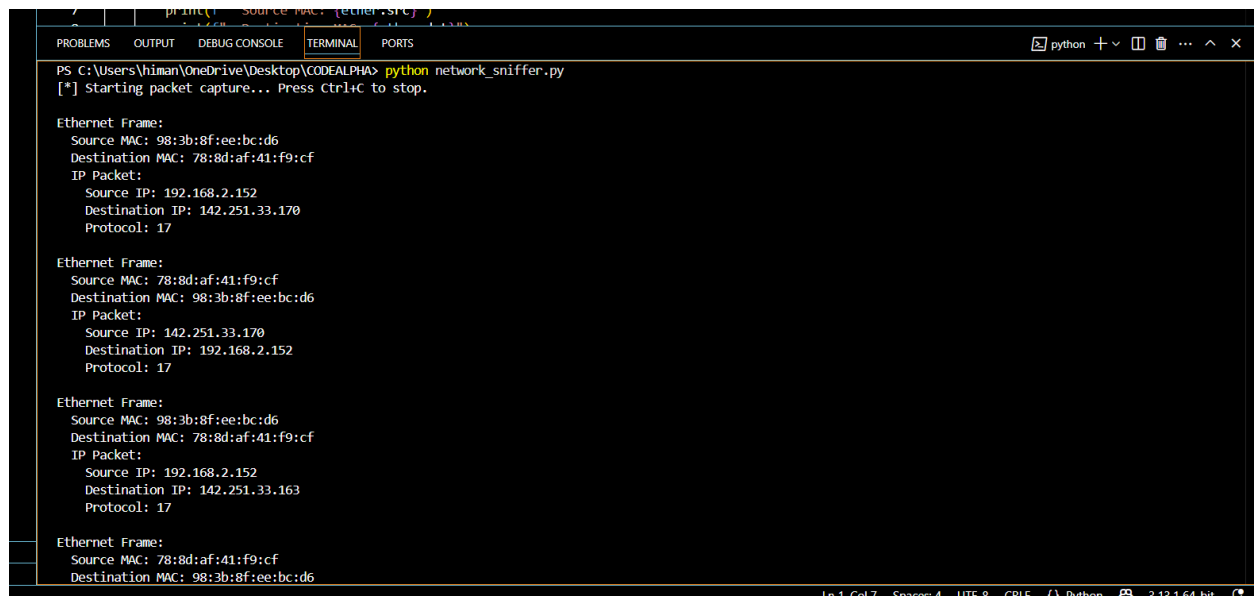
The screenshot shows the VS Code editor with the file explorer on the left displaying 'CODEALPHA', 'network_sniffer.py', and 'README.md'. The main editor area shows the 'network_sniffer.py' script. The script imports 'sniff', 'Ether', 'IP', and 'TCP' from 'scapy.all'. It defines a 'packet_callback' function that checks for 'Ether', 'IP', and 'TCP' layers in a packet and prints their details. The script then starts a packet capture using 'sniff' with the callback function and 'store=False'.

```
1 from scapy.all import sniff, Ether, IP, TCP
2
3 def packet_callback(packet):
4     if packet.haslayer(Ether):
5         ether = packet.getlayer(Ether)
6         print(f"\nEthernet Frame:")
7         print(f"    Source MAC: {ether.src}")
8         print(f"    Destination MAC: {ether.dst}")
9
10    if packet.haslayer(IP):
11        ip = packet.getlayer(IP)
12        print(f"    IP Packet:")
13        print(f"        Source IP: {ip.src}")
14        print(f"        Destination IP: {ip.dst}")
15        print(f"        Protocol: {ip.proto}")
16
17    if packet.haslayer(TCP):
18        tcp = packet.getlayer(TCP)
19        print(f"    TCP Segment:")
20        print(f"        Source Port: {tcp.sport}")
21        print(f"        Destination Port: {tcp.dport}")
22
23 print("[*] Starting packet capture... Press Ctrl+C to stop.")
24 sniff(prn=packet_callback, store=False)
25
```

3. Run the Script

1. Open the **terminal in VS Code**: Terminal → New Terminal
2. Run it as Administrator (important for packet sniffing)
3. Type: `python network_sniffer.py`

You'll start seeing live traffic like:



The screenshot shows the VS Code terminal window with the command prompt 'PS C:\Users\himan\OneDrive\Desktop\CODEALPHA> python network_sniffer.py'. The output shows the script starting packet capture and displaying live traffic details for three packets. Each packet is an Ethernet Frame containing an IP Packet with Protocol 17 (ICMP).

```
PS C:\Users\himan\OneDrive\Desktop\CODEALPHA> python network_sniffer.py
[*] Starting packet capture... Press Ctrl+C to stop.

Ethernet Frame:
Source MAC: 98:3b:8f:ee:bc:d6
Destination MAC: 78:8d:af:41:f9:cf
IP Packet:
Source IP: 192.168.2.152
Destination IP: 142.251.33.170
Protocol: 17

Ethernet Frame:
Source MAC: 78:8d:af:41:f9:cf
Destination MAC: 98:3b:8f:ee:bc:d6
IP Packet:
Source IP: 142.251.33.170
Destination IP: 192.168.2.152
Protocol: 17

Ethernet Frame:
Source MAC: 98:3b:8f:ee:bc:d6
Destination MAC: 78:8d:af:41:f9:cf
IP Packet:
Source IP: 192.168.2.152
Destination IP: 142.251.33.163
Protocol: 17

Ethernet Frame:
Source MAC: 78:8d:af:41:f9:cf
Destination MAC: 98:3b:8f:ee:bc:d6
```

```
17 | if packet.haslayer(TCP):
18 |     tcp = packet.getlayer(TCP)
19 |     print(f"    TCP Segment:")
20 |     print(f"        Source Port: {tcp.sport}")
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Source MAC: 98:3b:8f:ee:bc:d6
Destination MAC: 78:8d:af:41:f9:cf
IP Packet:
Source IP: 192.168.2.152
Destination IP: 142.251.33.163
Protocol: 17

Ethernet Frame:
Source MAC: 98:3b:8f:ee:bc:d6
Destination MAC: 78:8d:af:41:f9:cf
IP Packet:
Source IP: 192.168.2.152
Destination IP: 142.251.33.163
Protocol: 17

Starting packet capture... Press Ctrl+C to stop.

Ethernet Frame:

Source MAC: ...

Destination MAC: ...

IP Packet:

Source IP: 192.168.1.2

Destination IP: 8.8.8.8

Protocol: 6

TCP Segment:

Source Port: 56743

Destination Port: 443

Here you go