

Code Alpha Cybersecurity Internship

Submitted by:

Saira Arshad

Internship Period:

August 2025

Institution/Platform:

Code Alpha

Date of Submission:

August 2025

Table of Contents

1. Project Objective-----	
2. Tools Used-----	
3. Code Explanation-----	
4. Screenshots-----	
5. Output Samples-----	
6. Learnings and Challenges-----	
7. Conclusion-----	
8. Submission Info-----	

1. Project Objective

The objective of this task was to develop a basic network sniffer using Python. The program captures and analyzes network traffic in real time, identifies key protocols (TCP, UDP, ICMP), and extracts useful information such as source/destination IP addresses, ports, and payload data. This task aims to build foundational skills in packet analysis, protocol structure, and low-level networking.

2. Tools Used

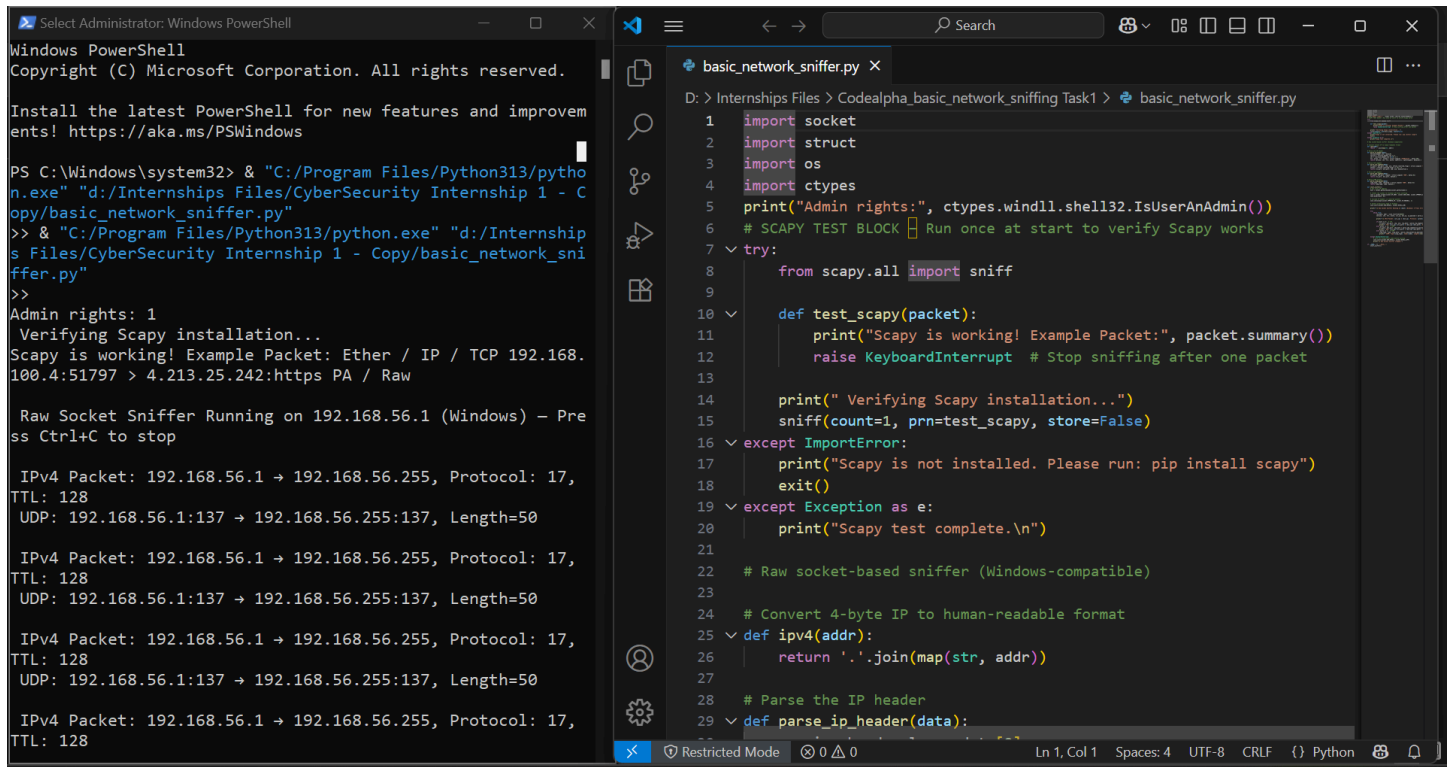
- - Python 3.10+: Core programming language used for development
- - Scapy: For packet manipulation and verification
- - socket: Used to create raw sockets for packet capturing
- - struct: Used to unpack binary data from network packets
- - PowerShell/CMD: Used to run the script with administrator privileges
- - Npcap: Windows packet capture driver required for raw sockets

3. Code Explanation

The script begins by checking if it is being run as an Administrator. It uses the socket module to create a raw socket that listens to all network traffic on the host machine. The IP header is parsed using struct.unpack to extract details such as TTL, protocol type, and IP addresses.

Depending on the protocol (TCP, UDP, ICMP), further functions are used to parse and display relevant information such as port numbers, sequence numbers, and data lengths. A Scapy block is also included at the beginning to verify that packet analysis tools are working correctly.

4. Screenshots



The screenshot shows a Windows PowerShell terminal window on the left and a VS Code editor on the right. The PowerShell window displays the execution of a Python script named `basic_network_sniffer.py`. The script checks for administrative rights, verifies Scapy installation, and then runs a raw socket-based sniffer. The VS Code editor shows the source code of `basic_network_sniffer.py`, which includes imports for `socket`, `struct`, `os`, and `ctypes`. It defines a `test_scapy` function to verify Scapy installation and a `sniff` function to capture network packets. The script also includes a `parse_ip_header` function to parse the IP header of captured packets.

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

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

PS C:\Windows\system32> & "C:/Program Files/Python313/python.exe" "d:/Internships Files/CyberSecurity Internship 1 - Copy/basic_network_sniffer.py"
>> & "C:/Program Files/Python313/python.exe" "d:/Internships Files/CyberSecurity Internship 1 - Copy/basic_network_sniffer.py"
>>
Admin rights: 1
Verifying Scapy installation...
Scapy is working! Example Packet: Ether / IP / TCP 192.168.100.4:51797 > 4.213.25.242:https PA / Raw

Raw Socket Sniffer Running on 192.168.56.1 (Windows) - Press Ctrl+C to stop

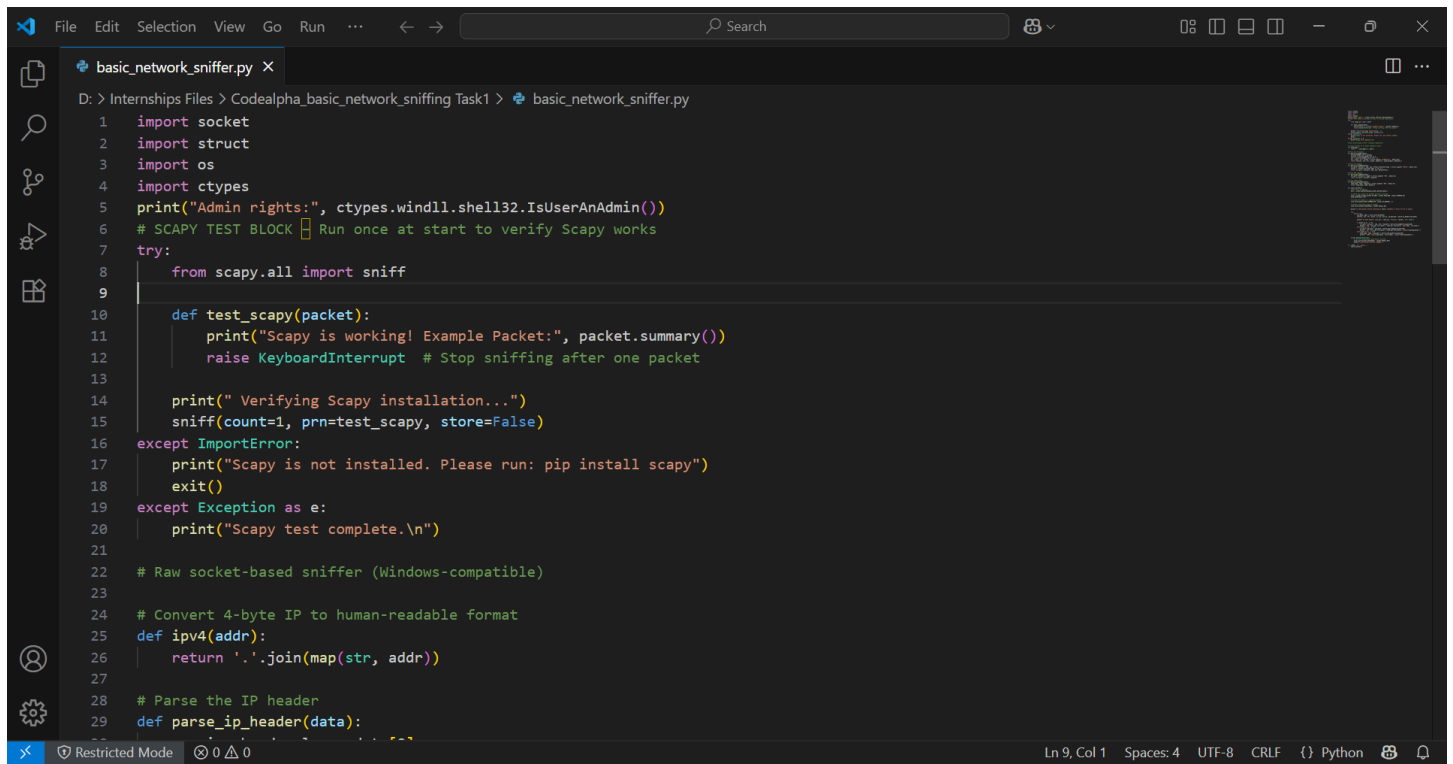
IPv4 Packet: 192.168.56.1 → 192.168.56.255, Protocol: 17, TTL: 128
UDP: 192.168.56.1:137 → 192.168.56.255:137, Length=50

IPv4 Packet: 192.168.56.1 → 192.168.56.255, Protocol: 17, TTL: 128
UDP: 192.168.56.1:137 → 192.168.56.255:137, Length=50

IPv4 Packet: 192.168.56.1 → 192.168.56.255, Protocol: 17, TTL: 128
UDP: 192.168.56.1:137 → 192.168.56.255:137, Length=50

IPv4 Packet: 192.168.56.1 → 192.168.56.255, Protocol: 17, TTL: 128
UDP: 192.168.56.1:137 → 192.168.56.255:137, Length=50
```

```
basic_network_sniffer.py
D: > Internships Files > Codealpha_basic_network_sniffing Task1 > basic_network_sniffer.py
1 import socket
2 import struct
3 import os
4 import ctypes
5 print("Admin rights:", ctypes.windll.shell32.IsUserAnAdmin())
6 # SCAPY TEST BLOCK Run once at start to verify Scapy works
7 try:
8     from scapy.all import sniff
9
10 def test_scapy(packet):
11     print("Scapy is working! Example Packet:", packet.summary())
12     raise KeyboardInterrupt # Stop sniffing after one packet
13
14 print("Verifying Scapy installation...")
15 sniff(count=1, prn=test_scapy, store=False)
16 except ImportError:
17     print("Scapy is not installed. Please run: pip install scapy")
18     exit()
19 except Exception as e:
20     print("Scapy test complete.\n")
21
22 # Raw socket-based sniffer (Windows-compatible)
23
24 # Convert 4-byte IP to human-readable format
25 def ipv4(addr):
26     return '.'.join(map(str, addr))
27
28 # Parse the IP header
29 def parse_ip_header(data):
```



The screenshot shows a VS Code editor window displaying the source code of the `basic_network_sniffer.py` script. The code is identical to the one shown in the previous screenshot, including imports, administrative rights check, Scapy verification, and the raw socket-based sniffer implementation.

```
basic_network_sniffer.py
D: > Internships Files > Codealpha_basic_network_sniffing Task1 > basic_network_sniffer.py
1 import socket
2 import struct
3 import os
4 import ctypes
5 print("Admin rights:", ctypes.windll.shell32.IsUserAnAdmin())
6 # SCAPY TEST BLOCK Run once at start to verify Scapy works
7 try:
8     from scapy.all import sniff
9
10 def test_scapy(packet):
11     print("Scapy is working! Example Packet:", packet.summary())
12     raise KeyboardInterrupt # Stop sniffing after one packet
13
14 print("Verifying Scapy installation...")
15 sniff(count=1, prn=test_scapy, store=False)
16 except ImportError:
17     print("Scapy is not installed. Please run: pip install scapy")
18     exit()
19 except Exception as e:
20     print("Scapy test complete.\n")
21
22 # Raw socket-based sniffer (Windows-compatible)
23
24 # Convert 4-byte IP to human-readable format
25 def ipv4(addr):
26     return '.'.join(map(str, addr))
27
28 # Parse the IP header
29 def parse_ip_header(data):
```

5. Output Samples

Admin rights: 1

Sniffing on 192.168.100.82 (Windows) — Press Ctrl + C to stop

IPv4 Packet: 192.168.100.82 → 239.255.255.250, Protocol: 17, TTL: 2

UDP: 192.168.100.82:55033 → 239.255.255.250:1900, Length=201

IPv4 Packet: 192.168.100.82 → 8.8.8.8, Protocol: 6, TTL: 64

TCP: 192.168.100.82:50321 → 8.8.8.8:443, Seq=123456789, Ack=987654321

6. Learnings and Challenges

Learned how raw sockets work and how to access low-level network data.

Understood how to parse IP, TCP, UDP, and ICMP headers using struct.

Faced issues with Windows permissions (WinError 10013) which required running the script as Administrator.

Gained hands-on experience with the Scapy library and its role in cybersecurity tools.

7. Conclusion

This project successfully met the requirements of Task 1 in the Code Alpha Cybersecurity Internship. It provided practical experience with raw sockets, packet parsing, and network protocol structures. The ability to interpret and analyze network packets is a valuable skill in the field of cybersecurity.

8. Submission Info

Name: Saira Malik

Internship: Code Alpha Cybersecurity Internship

Task: Task 1 - Basic Network Sniffer

Date of Submission: August 2025