Network Sniffer Project Report

Project Name: Basic Network Sniffer
Internship Program: CodeAlpha Cyber Security Internship
Intern Name: Mahnoor Shafi
Intern Email: mahnoorshafi88@gmail.com
Date: [20-8-2025 - 20-9-2025]

Table of Contents:

1-Introduction

2-Tools & Requirements

3-Project Objective

4-Implementation Steps

- 4.1 Installing Dependencies
- 4.2 Writing the Sniffer Script
- 4.3 Running the Sniffer

5-Advantages of a Sniffer

6-Real-World Applications

7-Conclusion

1. Introduction

A **network sniffer** is a tool that captures and analyzes network traffic (packets) in real time.
In this project, I created a **Python-based sniffer** using the **Scapy** library to understand how data flows across networks and to analyze packet details.

2. Tools & Requirements

Python 3.10+

Scapy library (pip install scapy)

Npcap (for Windows packet capturing)

Command Prompt / Terminal

(Optional) **Wireshark** for deeper packet analysis

3. Project Objective

The main objective was to:

Capture live network packets.

Display **Source IP, Destination IP, and Protocol**.

Save captured packets into a .pcap file for analysis in Wireshark.

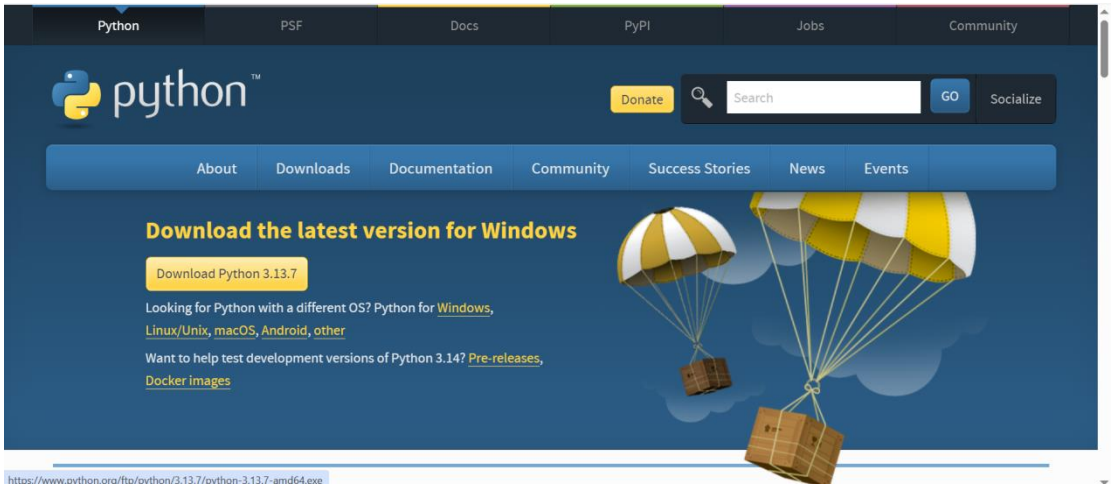
4. Implementation Steps

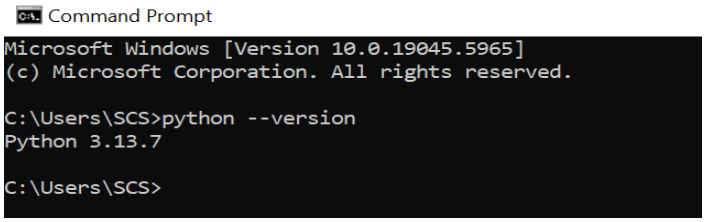
4.1 Installing Dependencies

Installed Python and added to PATH.

Installed **Npcap** for packet capturing.

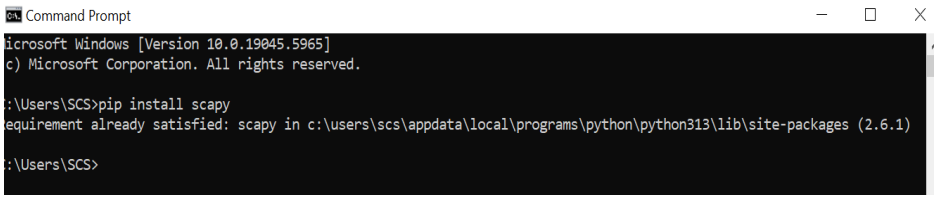
Installed **Scapy** library using pip install scapy.





Npcap:

Scapy:

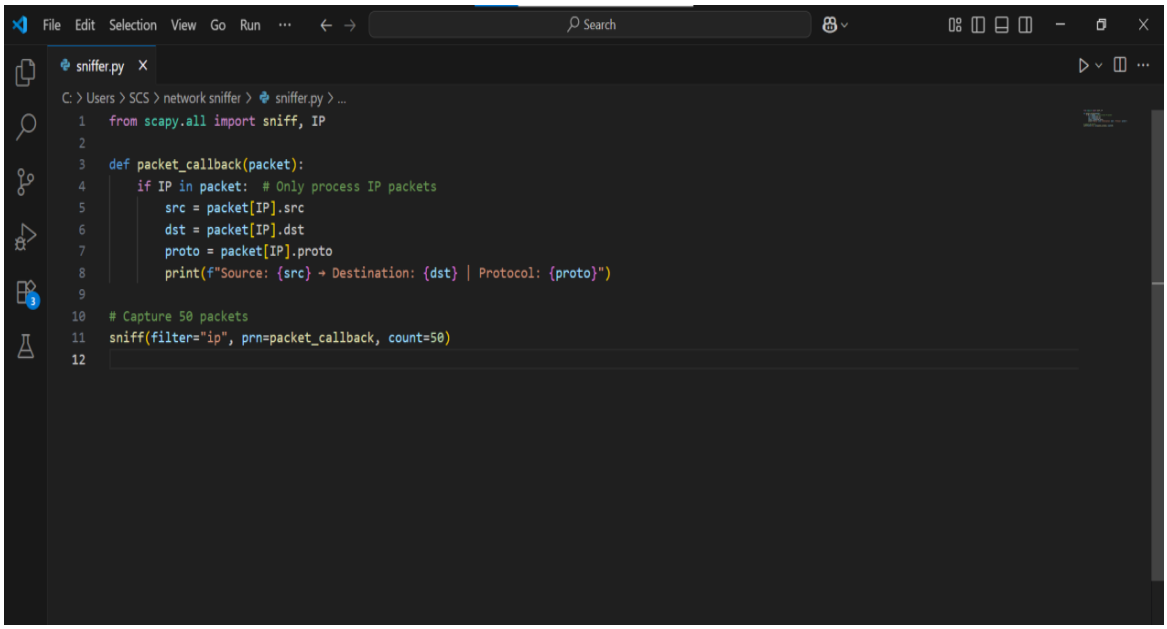


4.2 Writing the Sniffer Script

Basic code written in Python:

```
from scapy.all import sniff, IP
def packet_callback(packet):
    if IP in packet:
        src = packet[IP].src
        dst = packet[IP].dst
        proto = packet[IP].proto
        print(f"Source: {src} → Destination: {dst} | Protocol: {proto}")

sniff(filter="ip", prn=packet_callback, count=50)
```



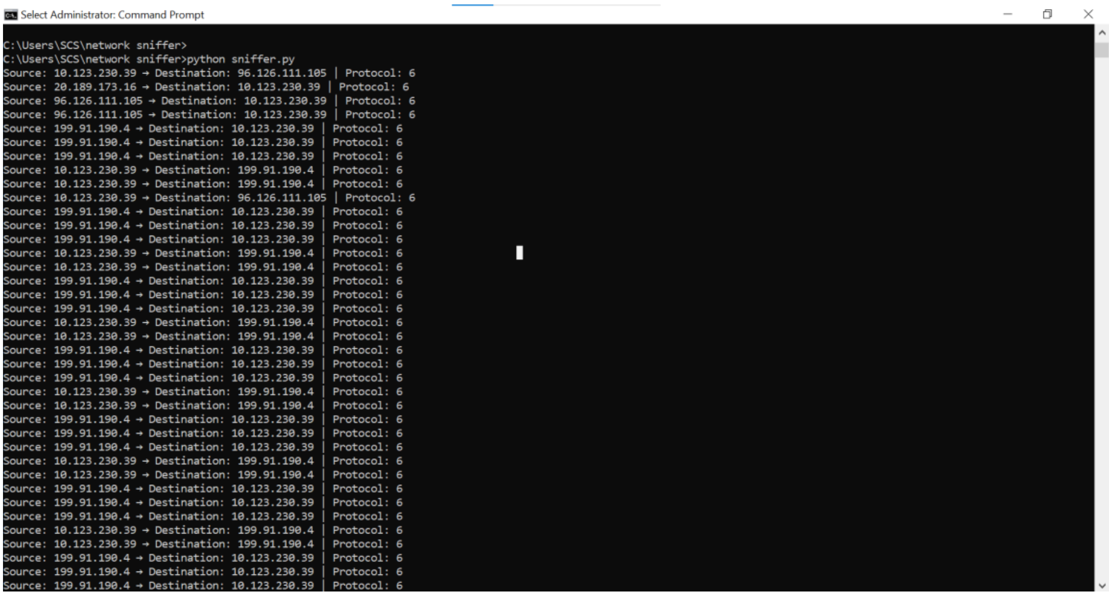
4.3 Running the Sniffer

Opened Command Prompt.

Navigated to project folder.

Ran the script:

python sniffer.py



Source: 10.123.230.39 → Destination: 199.91.190.4 | Protocol: 6
Source: 10.123.230.39 → Destination: 8.8.8.8 | Protocol: 1
Protocol: 6 → TCP

Protocol: 17 → UDP

Protocol: 1 → ICMP (ping)

5. Advantages of a Sniffer

Helps understand how network traffic works.

Useful for **cybersecurity investigations** and **digital forensics**.

Detects unusual or malicious activity.

6. Real-World Applications

Used in **Security Operations Centers (SOC)**.

Network troubleshooting and monitoring.

Evidence collection during **forensic investigations**.

Understanding malware communication patterns.

7. Conclusion

The Python-based network sniffer successfully demonstrated packet capturing and analysis in real time.
It provided insights into network communication and built the foundation for more advanced cybersecurity projects like **Intrusion Detection Systems**.