

1. Title: Basic Network Sniffer – CodeAlpha Internship Task

Intern Name: **Hardik Rathod**

Domain: **Cyber Security**

Internship Program: **CodeAlpha Virtual Internship**

Task Completed: **Task 1 – Basic Network Sniffer**

GitHub Repo: https://github.com/HCS-9/CodeAlpha_Basic-Network-Sniffer-

LinkedIn Post: [linkedin.com/in/hardik-rathod-7a9257361](https://www.linkedin.com/in/hardik-rathod-7a9257361)

Submission Date: **[24-07-2025]**

2. Introduction

This project is a basic network sniffer written in Python using the Scapy library. It captures real-time network packets and displays useful information like source and destination IPs, ports, protocols, and payloads. The goal is to understand how data flows through the network and analyze the packet structure.

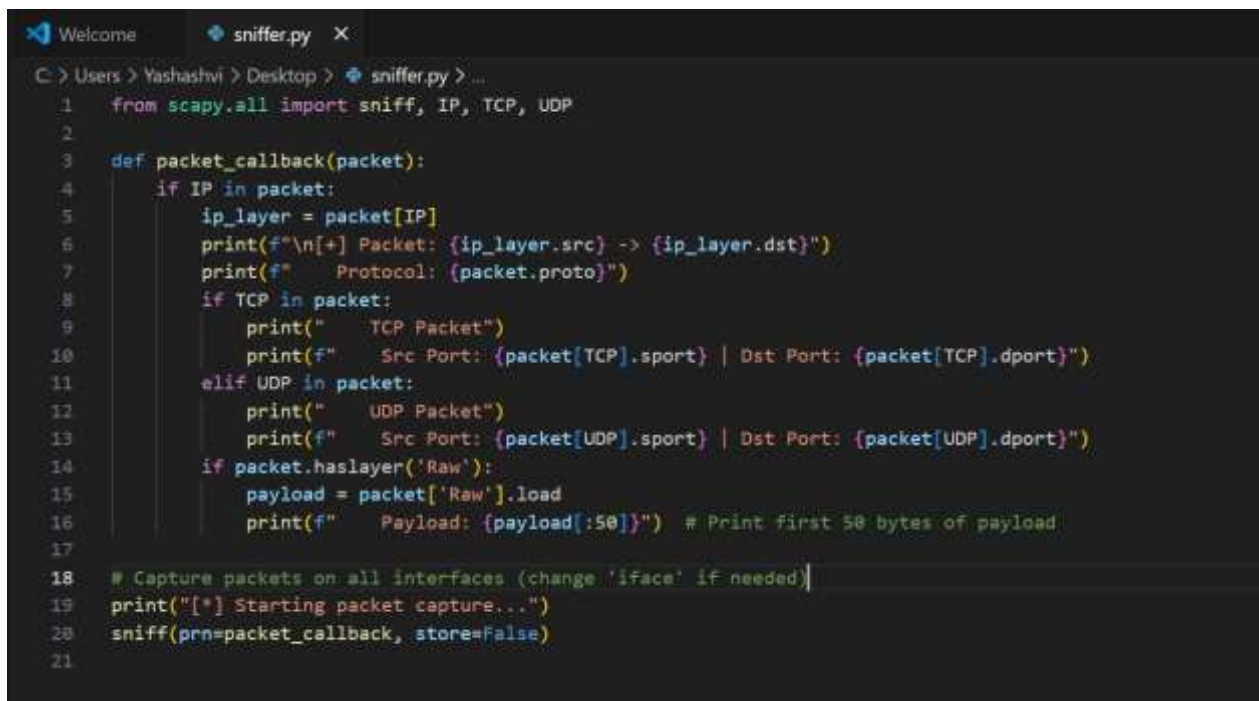
3. Tools and Technologies Used

- **Python 3**
- **Scapy library**
- **Terminal / Command Prompt**
- **GitHub**
- **(Optional) curl or browser for generating traffic**

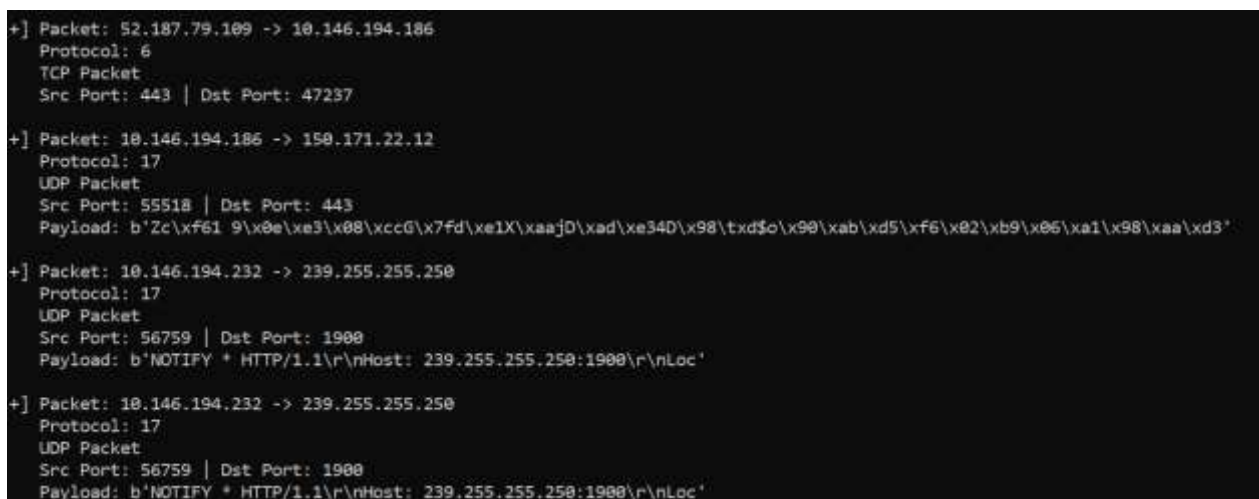
4. How the Project Works – Step-by-Step

1. The user runs the Python script with administrator/root permissions.
2. The script uses Scapy to sniff live network traffic on port 80 (HTTP).
3. For every captured packet:
 - The source and destination IPs are printed.
 - The source and destination ports are printed.
 - If payload data is present, it is extracted and displayed.
4. The user can interact with the browser or terminal to generate HTTP traffic (e.g., by visiting <http://example.com>).
5. The script captures and displays the network activity in real-time.

5. Screenshots



```
1 from scapy.all import sniff, IP, TCP, UDP
2
3 def packet_callback(packet):
4     if IP in packet:
5         ip_layer = packet[IP]
6         print(f"\n[+] Packet: {ip_layer.src} -> {ip_layer.dst}")
7         print(f"    Protocol: {packet.proto}")
8         if TCP in packet:
9             print("    TCP Packet")
10            print(f"    Src Port: {packet[TCP].sport} | Dst Port: {packet[TCP].dport}")
11        elif UDP in packet:
12            print("    UDP Packet")
13            print(f"    Src Port: {packet[UDP].sport} | Dst Port: {packet[UDP].dport}")
14        if packet.haslayer('Raw'):
15            payload = packet['Raw'].load
16            print(f"    Payload: {payload[:50]}") # Print first 50 bytes of payload
17
18 # Capture packets on all interfaces (change 'iface' if needed)
19 print("[*] Starting packet capture...")
20 sniff(prn=packet_callback, store=False)
21
```



```
+ ] Packet: 52.187.79.109 -> 10.146.194.186
Protocol: 6
TCP Packet
Src Port: 443 | Dst Port: 47237

+ ] Packet: 10.146.194.186 -> 150.171.22.12
Protocol: 17
UDP Packet
Src Port: 55518 | Dst Port: 443
Payload: b'Zc\xf61 9\xe0\xe3\xe8\xccG\x7fd\xe1X\xaaJD\xad\xe34D\x98\txd$0\x90\xab\xd5\xf6\xe2\xb9\xe6\xa1\x98\xaa\xd3'

+ ] Packet: 10.146.194.232 -> 239.255.255.250
Protocol: 17
UDP Packet
Src Port: 56759 | Dst Port: 1900
Payload: b'NOTIFY * HTTP/1.1\r\nHost: 239.255.255.250:1900\r\nLoc'

+ ] Packet: 10.146.194.232 -> 239.255.255.250
Protocol: 17
UDP Packet
Src Port: 56759 | Dst Port: 1900
Payload: b'NOTIFY * HTTP/1.1\r\nHost: 239.255.255.250:1900\r\nLoc'
```