CodeAlpha Cyber Security Internship

Project 1: Basic Network Sniffer

Submitted by: [Your Name]

1. Introduction

A network sniffer is a tool that captures and analyzes packets traveling through a network. It helps security professionals understand how data flows between devices, identify potential threats,
and debug network issues. This project demonstrates the creation of a basic packet sniffer in Python using the scapy library.

2. Objectives

- Capture live network packets

- Extract useful information such as source and destination IPs, protocol, and payload

- Learn basics of packet structures and protocols

- Save captured packets for later analysis in tools like Wireshark

3. Methodology

Step 1: Install the required Python libraries (scapy).

Step 2: Write a Python program that uses sniff() function from scapy to capture packets.

Step 3: Process each packet using a callback function and display a summary.

Step 4: Optionally, save captured packets to a .pcap file for analysis in Wireshark.

Step 5: Test the program by capturing network traffic in real-time.

4. Code Implementation

Basic Network Sniffer

```
from scapy.all import sniff
```

```
# Function to process each packet
def packet_callback(packet):
   print(packet.summary())   # Prints short info of packet

# Capture packets
print("Starting Packet Sniffer... Press CTRL+C to stop.")
sniff(prn=packet_callback, count=20)   # capture 20 packets
```

Additional Features

Filter by Protocol (e.g., TCP Only):

```
sniff(filter="tcp", prn=packet_callback, count=10)
```

Save Packets for Wireshark Analysis:

```
from scapy.all import wrpcap
packets = sniff(count=50)
wrpcap("captured_packets.pcap", packets)
```

Continuous Live Capture:

```
sniff(prn=packet_callback, store=False)
```

5. Sample Output

Example of captured packets:

```
Ether / IP / TCP 192.168.1.5:49832 > 142.250.182.14:https S

Ether / IP / UDP 192.168.1.5:56892 > 8.8.8.8:domain

Ether / IP / ICMP 192.168.1.5 > 192.168.1.1 echo-request
```

6. Conclusion

The Basic Network Sniffer project demonstrates how Python can be used to capture and analyze network packets. This task provides practical exposure to packet structures, network monitoring, and the basics of intrusion detection.

The implementation can be further extended to include filtering, alerting, and integration with security monitoring systems.