CodeAlpha Cybersecurity Internship – Task 1 Report

Project Title: Basic Network Sniffer in Python

Intern Name: Sajid Nazir

Date: [20-07-2025]

# 1. Introduction

As part of the CodeAlpha Cybersecurity Internship, I selected Task 1: Building a Basic Network Sniffer. This project involved writing a Python program using the Scapy library to capture, analyze, and display live network traffic packets in Kali Linux.

# 2. Objectives

The key objectives of this task were: Understand how data flows through a network.- Capture and analyze network packets.- Display useful details like source/destination IPs, protocols, and payloads.- Gain hands-on experience with packet sniffing and Python networking libraries. The objective of this task was to develop a simple network packet sniffer using Python and Scapy that captures and displays real-time packet-level data. The script displays crucial information such as IP addresses, protocols, and payloads. This helps to understand how network traffic works, how to monitor it effectively, and provides hands-on experience with packet-level inspection.

# 3. Tools and Technologies Used

|  |  |
| --- | --- |
| Tool | Purpose |
| Kali Linux | Operating system for development |
| Python 3 | Programming language |
| Scapy | Library to capture and analyze packets |
| Terminal | Run and test the sniffer |

# 4. Code Explanation

The Python script uses Scapy to sniff packets and extract information. The key parts include:  
- Importing necessary modules from scapy.  
- Defining a packet\_callback function to analyze IP, TCP, and UDP packets.  
- Printing source IP, destination IP, protocol number, and payload.  
- Using sniff() to start live capture.

Sample Code:

from scapy.all import sniff, IP, TCP, UDP  
  
def packet\_callback(packet):  
 if IP in packet:  
 ip\_src = packet[IP].src  
 ip\_dst = packet[IP].dst  
 proto = packet[IP].proto  
 print(f"[+] Source: {ip\_src} -> Destination: {ip\_dst} | Protocol: {proto}")  
 if TCP in packet:  
 print("TCP Payload:", bytes(packet[TCP].payload))  
 elif UDP in packet:  
 print("UDP Payload:", bytes(packet[UDP].payload))  
 print("=" \* 60)  
  
print("🔎 Sniffing Started... Press Ctrl+C to stop.")  
sniff(prn=packet\_callback, store=0)

# 5. Screenshots

# Insert your screenshots here using Word or LibreOffice. For example:

# 

Figure 1: Output of the sniffer script showing captured packets

# 

# 🛠 Setup & Installation

# Step 1: Update system packages and install pip3

# ```bash

# sudo apt update

# sudo apt install python3-pip

# Step 2: Install Scapy

# bash

# pipx install scapy

# # OR use a virtual environment (recommended)

# python3 -m venv venv

# source venv/bin/activate

# pip install scapy

# 6. Learning Outcomes

- Understood basics of packet sniffing and network protocols.  
- Learned to use Scapy to interact with low-level network traffic.  
- Learned to run Python scripts with root privileges in Kali Linux.  
- Gained practical experience building cybersecurity tools.

# 7. GitHub Repository Link:

# <https://github.com/sajidmir-m/NetworkSniffer-CodeAlpha.git>

# 8. Conclusion

This task provided valuable hands-on exposure to networking and security concepts. It helped me understand how real-world tools like Wireshark or tcpdump work under the hood. I am thankful to CodeAlpha for this learning opportunity.