Python Packet Sniffer Project Report

Project Title:

Python-Based Network Packet Sniffer using Scapy

Developed By:

Name: Humair Ali

Internship: Cybersecurity Internship

Date: June 2025

Description:

This project is a Python-based network packet sniffer built using the Scapy library on Ubuntu Linux. The goal was to develop a tool that can capture and analyze real-time network traffic, helping understand protocol structures and data flow across the network.

Objectives Achieved:

- Build a Python program to capture live packets
- Analyze packet structure and headers
- Understand data flow and networking protocols
- Use Scapy for packet sniffing and analysis
- Display useful info: source IP, destination IP, protocols, and payload

Tools Used:

- Python 3
- Scapy Library
- Ubuntu Linux (Virtual Machine)

Python Packet Sniffer Project Report

- Terminal / CLI

1. **Install Scapy:**

sudo apt update sudo

apt install python3-pip

pip install scapy

Run the Sniffer:

sudo python3 sniffer.py

Note: Root privileges (sudo) are required to sniff packets.

Sample Code Overview (sniffer.py):

from scapy.all import sniff, IP, TCP, UDP

def packet_callback(packet):

if packet.haslayer(IP): ip_layer = packet[IP] print(f"[+] From: {ip_layer.src} --> To: {ip_layer.dst} | Protocol: {ip_layer.proto}") if packet.haslayer(TCP) or packet.haslayer(UDP):

print("Payload:", bytes(packet.payload)) print("-" * 50)

sniff(prn=packet_callback, count=20)

Code Image:

Python Packet Sniffer Project Report

Open sniffer.py

```
from scapy.all import sniff, IP, TCP, UDP, ICMP, Raw, DNS, DNSQR
from datetime import datetime

def analyze_packet(packet):
    print("\n" + "-"*60)
    print("Packet Captured at:", datetime.now().strftime("%Y-%m-%d %

    if IP in packet:
        ip_layer=packet[IP]
        print(f"Source IP: {ip_layer.src}")
        print(f"Destination IP: {ip_layer.dst}")

    if packet.haslayer(DNS) and packet.getlayer(DNS).qr==0:
        dns_layer=packet.getlayer(DNS)
        query_name=dns_layer.qd.qname.decode('utf-8')
        print(f"DNS Query: {query_name}")

    if packet.haslayer(TCP) and  packet.haslayer(Raw):
        try:
            payload=packet[Raw].load.decode('utf-8', errors='ignore')

            if payload.startswith("GET") or payload.startswith("POST"
                print("HTTP Request Detected:")
                print(payload.split("\r\n")[0])
                for line in payload.split("\r\n")[1:]:
                    if line=="":
                        break
```

Sample Output:

[+] From: 192.168.0.105 --> To: 142.250.182.42 | Protocol: 6

Payload: b'...HTTP Data...'

Python Packet Sniffer Project Report

[+] From: 192.168.0.105 --> To: 192.168.0.1 | Protocol: 17

Payload: b'...DNS Data...'

Files Included:

PacketSnifferProject/ sniffer.py	Main Python script
Sniffer Output Proof.jpg	Output from running the script
Report.txt	This documentation

Contact:

For any queries related to this project, feel free to contact me:

Email: humairali2612@gmail.com

LinkedIn: https://www.linkedin.com/in/humair-ali-a6b3a7339/