Network Packet Sniffer with Alert System

Introduction

This project, **Network Packet Sniffer with Alert System**, was developed as part of a cybersecurity internship. The tool captures live packets, logs traffic details, detects anomalies such as flooding or port scanning, and visualizes data. It simulates the role of a SOC (Security Operations Center) analyst by providing real-time monitoring, detection, and alerting.

Abstract

The project implements a Python-based packet sniffer using Scapy to capture network packets in real time. Each packet’s metadata (timestamp, source IP, destination IP, protocol, and size) is stored in an SQLite database. An anomaly detection rule identifies IP addresses exceeding traffic thresholds, triggering alerts to indicate potential malicious activity. Additionally, Matplotlib is used to generate graphs showing protocol distribution, combining monitoring, detection, and reporting into one system.

Tools Used

- Python: Programming language for implementation
- Scapy: Packet sniffing and manipulation
- SQLite: Structured log storage
- Matplotlib: Graph visualization
- SMTP: Sending email alerts

Steps Involved in Building the Project

1. Installed Python and required dependencies.
2. Configured Scapy to sniff packets from the network interface.
3. Extracted key details (source IP, destination IP, protocol, size).
4. Logged details into an SQLite database.
5. Applied threshold rule (>50 packets from the same IP = suspicious).
6. Triggered alerts when suspicious activity was detected.
7. Visualized packet distribution across protocols using Matplotlib.

Conclusion

The Network Packet Sniffer with Alert System successfully simulates the initial role of a SOC analyst. It provides hands-on experience with packet capture, traffic analysis, anomaly detection, and alerting. This project not only strengthens cybersecurity fundamentals but also demonstrates how monitoring and visualization aid in identifying potential threats.