Network Packet Sniffer with Alert System

1. Introduction

With the rapid increase in cyberattacks such as port scanning, denial-of-service (DoS), and unauthorized access attempts, monitoring network traffic has become essential for cybersecurity. Traditional packet sniffers like Wireshark provide detailed packet analysis, but they lack an automated alert mechanism for suspicious activity.

This project implements a **real-time network packet sniffer with an integrated alert system** to monitor traffic, detect anomalies, and generate alerts for potentially malicious behavior. The system helps security analysts identify threats at an early stage, improving overall network defense.

2. Abstract

The Network Packet Sniffer with Alert System is a Python-based tool that leverages the **Scapy** library to capture and analyze live packets. The system monitors TCP, UDP, and ICMP traffic and applies detection logic to identify suspicious patterns, such as:

- **Port Scans** → when one IP probes multiple ports in a short time
- **SYN Floods** → excessive SYN packets from a single source
- **ICMP Floods** → abnormally high ping requests

When such behavior is detected, the system logs the event in alerts.log and displays warnings in the terminal.

This project demonstrates how network monitoring and intrusion detection can be achieved with open-source tools and simple detection algorithms.

3. Tools & Technologies Used

- **Programming Language**: Python 3
- **Libraries**: Scapy, SQLite (optional for logs), time
- **Platform**: Kali Linux (but works on any Linux/Windows system with Scapy)
- **Output**: Real-time alerts in terminal + log file

4. Methodology

1. **Packet Capture** → Used Scapy's sniff() function to capture live packets.

2. **Packet Analysis** → Extracted source IP, destination IP, protocol, and flags.
3. **Detection Logic**:
    - Flagged multiple SYN packets from one IP as potential **SYN Flood**
    - Flagged sequential connections from one IP to many ports as **Port Scan**
    - Flagged excessive ICMP echo requests as **ICMP Flood**
4. **Alert System** → Suspicious activity was logged in alerts.log with timestamp and details.
5. **Testing** → Simulated port scan using nmap and DoS attempts to validate detection.

5. Results

- Successfully captured live network packets.
- Detected and logged suspicious activities such as **Nmap scans** and repeated SYN requests.
- Generated real-time alerts in terminal for immediate action.
- Verified logs were stored persistently in alerts.log.

6. Conclusion

The **Network Packet Sniffer with Alert System** provides a lightweight Intrusion Detection System (IDS) using open-source tools. While not as advanced as enterprise IDS solutions, it demonstrates the core concepts of:

- **Packet sniffing**
- **Traffic pattern recognition**
- **Alert generation for suspicious activity**

Future enhancements may include:

- Integration with a GUI dashboard for visualization
- Advanced ML-based anomaly detection
- Email/SMS notifications for critical alerts

This project showcases practical cybersecurity skills in **network monitoring, intrusion detection, and security automation**, making it an excellent foundation for real-world security systems.