# Network Packet Sniffer with Alert System

## Abstract

This project implements a real-time Network Packet Sniffer with an alert system to detect anomalies in network traffic. Using Python and Scapy, it captures live packets, stores essential headers in an SQLite database, and applies anomaly detection mechanisms such as port scanning and flooding detection. The system also provides alerts when suspicious activity exceeds defined thresholds, ensuring network security monitoring in real time.

## Introduction

In today’s digital world, network security is of utmost importance. Monitoring real-time traffic helps detect potential threats early. A packet sniffer is a tool that captures and analyzes network packets. By combining it with anomaly detection and alerting, the project strengthens proactive network defense.

## Tools Used

1. Python – Programming language for implementation
2. Scapy – For packet capturing and analysis
3. SQLite – For storing packet logs
4. Matplotlib – For traffic visualization
5. Threading – For real-time capture and logging

## Steps Involved in Building the Project

1. Capturing packets using Scapy and extracting headers (IP, port, length, flags).
2. Storing packet data into an SQLite database for persistence.
3. Implementing anomaly detection mechanisms such as port scanning and flooding.
4. Generating alerts when suspicious behavior crosses defined thresholds.
5. Visualizing traffic statistics using Matplotlib.
6. (Optional) Adding GUI for real-time monitoring.

## Conclusion

The project successfully demonstrates real-time packet sniffing, anomaly detection, and alerting using Python. It highlights the importance of monitoring and analyzing network traffic to identify potential security risks. This system can be extended with more advanced detection techniques, integration with SIEM tools, and improved GUI features for professional use.