# Real-Time Network Packet Sniffer with Alert System

## Introduction

In the era of increasing cyber threats, proactive network monitoring is crucial to detect and mitigate malicious activities. Packet sniffing is a foundational approach used to inspect, analyze, and log network traffic in real time. This project presents a comprehensive real-time network packet sniffer with built-in anomaly detection and a user-friendly graphical interface.

## Abstract

This project is a Python-based application that captures and analyzes network traffic in real-time using Scapy. It includes a Tkinter GUI with live visualizations and dashboards, anomaly detection for port scans and flooding, logging of DNS requests and credentials, and optional email alerts. The system leverages SQLite for persistent log storage and supports user-configurable settings, export options, and live traffic graphs categorized by protocol and port usage. It is intended as a lightweight yet powerful network security tool for education, research, or enterprise use.

## Tools Used

* **Python 3.10+**
* **Scapy** – for low-level network packet capturing
* **Tkinter + TTKBootstrap** – for GUI and modern UI theming
* **Matplotlib** – for real-time graph plotting
* **SQLite** – for storing packets, DNS logs, and credentials
* **Threading** – to ensure non-blocking UI updates
* **smtplib** – for sending anomaly alerts via email
* **dotenv** – for managing environment variables securely

## Steps Involved in Building the Project

1. Packet Sniffing Setup  
   Used Scapy to sniff live traffic and extract IP, port, DNS, and raw payload data.
2. Database Integration  
   Designed SQLite schemas to store:
   * All packet metadata (packet\_logs.db)
   * DNS queries (dns\_logs.db)
   * Extracted credentials (credentials.db)
3. Anomaly Detection Module  
   Implemented detection for:
   * **Port scans** (10+ different ports within 10s)
   * **Flood attacks** (100+ packets from same IP in 10s)  
     Alerts are sent to the configured email address using SMTP.
4. Graphical User Interface  
   Designed a GUI using Tkinter and ttkbootstrap featuring:
   * Live summary (packet count, protocol breakdown, anomaly count)
   * Real-time graphs for protocol or port usage (top 20 ports)
   * Logs section with filtering, sub-tabs, and export capability
   * Credential sniffing toggle and settings panel
5. Log Visualization  
   Integrated:
   * Real-time packet log preview
   * DNS request logs in tabular view
   * Credential logs showing potential leaks
   * Search & CSV export for each tab
6. Packaging & Extensions  
   Application can be packaged into an .exe using PyInstaller and optionally extended with:
   * Firewall rule suggestions
   * IP blocking from GUI
   * Slack/webhook integrations

## Conclusion

This network packet sniffer project demonstrates the ability to monitor and respond to real-time network anomalies with an intuitive user interface and lightweight backend. It is scalable and extendable for advanced intrusion detection, making it suitable for use in academic, home lab, or enterprise environments. The modular design ensures it can grow with evolving network security needs.