Project Report: Live Network Security Monitor

NAME : B.Hemanth **Date:** September 8, 2025

Introduction

In an era of increasing digital connectivity, network security has become a paramount concern for individuals and organizations alike. The ability to monitor network traffic in real-time is crucial for identifying unauthorized access, malicious activities, and potential security breaches. Standard security measures like firewalls are essential but often fall short of detecting novel or sophisticated threats. This project addresses the need for a more intelligent and responsive monitoring system by developing a Live Network Security Monitor. The objective is to create a high-performance tool that not only captures and logs network data but also uses machine learning to automatically detect anomalous patterns that could indicate a security threat.

Abstract

This report details the design and implementation of a real-time network security monitor developed in Python. The system is engineered to capture network packets efficiently, analyze them for anomalies using a machine learning model, and present the findings on an interactive web dashboard. The architecture employs a multi-threaded producer-consumer model to handle high traffic volumes without packet loss. Network data is captured using the Scapy library, while an Isolation Forest algorithm from Scikit-learn is used to distinguish between normal and anomalous traffic patterns. All captured data is persistently stored in a local SQLite database. The user interface is a live, web-based dashboard created with Streamlit, which visualizes key network metrics, including the latest captured packets, top source IP addresses, and protocol distribution, providing an intuitive and accessible overview of network health and security.

Tools Used

The project leverages a combination of powerful open-source Python libraries to achieve its functionality:

- **Python:** The core programming language used for the entire project, chosen for its extensive libraries and ease of development.
- **Scapy:** A powerful packet manipulation library used for sniffing, capturing, and dissecting network packets in real-time.
- **Scikit-learn:** A comprehensive machine learning library used to implement the Isolation Forest algorithm for anomaly detection.
- **Streamlit:** A modern web framework used to rapidly create and deploy the interactive, real-time data dashboard.
- **SQLite:** A self-contained, serverless SQL database engine used for lightweight, persistent logging of all captured packet data.
- **Pandas & NumPy:** Essential libraries used for data manipulation and numerical operations, particularly for preparing data for the machine learning model.

Steps Involved in Building the Project

The development of the project was carried out in a modular and sequential manner to ensure each component was robust before integration.

1. **Packet Capture and Architecture Design:** The initial step involved using Scapy to capture raw network packets. To ensure the system could handle high traffic loads, a multi-threaded producer-consumer architecture was designed. One thread is dedicated solely to capturing packets and placing them into a queue, while a separate consumer thread processes them. This decouples packet capture from analysis, preventing bottlenecks and packet loss.
2. **Database Integration:** A lightweight SQLite database was chosen for data persistence. A schema was designed to store essential packet information (timestamp, source/destination IP, protocol, ports, length). To optimize performance, a batch insertion method (`executemany`) was implemented, allowing the system to write data to the database in efficient chunks rather than one packet at a time.
3. **Machine Learning Model Development:** The core of the intelligent monitoring is the anomaly detection model.
    a. **Data Collection:** A dedicated script (`train_model.py`) was created to capture a baseline sample of "normal" network traffic.
    b. **Training:** An Isolation Forest algorithm was trained on this baseline data. This unsupervised model is highly effective at identifying outliers without needing pre-labeled malicious data.

    c. **Model Persistence:** The trained model was saved to a file (`anomaly_model.pkl`) using `joblib` for easy loading and use during live monitoring.

4. **Real-time Analysis and Integration:** The trained ML model was integrated into the packet processing pipeline. For each packet captured by the sniffer, its features are extracted and fed to the model for a prediction. Packets identified as anomalies are flagged for potential review.
5. **Dashboard Development:** The final step was to create a user-friendly interface. Streamlit was used to build a web dashboard that connects to the SQLite database. The dashboard automatically refreshes every few seconds to display the latest captured packets, charts of top source IPs and protocol distributions, and other relevant metrics, providing a live and intuitive view of network activity.

Conclusion

This project successfully demonstrates the creation of a modern, effective network security monitoring tool. By integrating high-performance design patterns with machine learning, the system provides capabilities beyond those of traditional packet sniffers. The resulting application is not only functional but also highly extensible. Future work could involve implementing more sophisticated detection models, adding an automated alerting system (e.g., via email or SMS), and expanding the dashboard to include more detailed historical analysis and threat reporting features. The project serves as a strong foundation for developing more advanced cybersecurity solutions.