IoT Network Anomaly Detection and Visualization

Mohamed Fouad Rabahi

July 27, 2025

Abstract

This project aims to detect anomalies in IoT network traffic by analyzing live or logged packets, extracting relevant features, applying rule-based anomaly detection, and visualizing the results on a web interface. It includes real-time traffic analysis, a feature extractor, and a Flask-based dashboard for reviewing anomalies over time.

1 Introduction

IoT devices are increasingly widespread, making them attractive targets for cyberattacks. This project provides a practical solution to monitor traffic, detect anomalies, and visualize those anomalies to enhance situational awareness and threat detection.

2 Project Structure

- **packet_analyzer.py**: Captures and logs network packets using Scapy.
- **detector.py**: Aggregates packets per second, extracts features, and labels anomalies.
- **feature_extractor.py**: (Optional) Extracts additional features like average packet size, TCP/UDP counts, etc.
- **web/app.py**: Flask app to visualize the CSV-based anomaly data.
- **static/** and **templates/**: Contain CSS and HTML templates for the web dashboard.

3 Functionalities

1. **Live Packet Capture**: Captures packets and logs them to a CSV file.
2. **Feature Extraction**: Aggregates features per second from captured packets.
3. **Anomaly Detection**: Rule-based flagging of suspicious behavior (e.g., high packet rate, high UDP count).
4. **Visualization Dashboard**:
    - Summary statistics
    - Real-time anomaly chart
    - Tabular view of labeled packets

4 Technologies Used

- Python (Scapy, Pandas, Flask)
- HTML, CSS, Bootstrap
- JavaScript (Chart.js for visualization)

5 Dataset

Two CSV files were used:

- `packet_log.csv`: Raw captured packets
- `labeled_traffic.csv`: Aggregated and labeled traffic data

6 Running the Project

To run this project, refer to the README for environment setup and instructions. The dashboard automatically updates upon CSV refresh.

7 Future Improvements

- Implement ML-based anomaly detection
- Add user authentication to the web app
- Improve data filtering (e.g., by protocol, port, etc.)

8 Conclusion

This project successfully combines real-time network monitoring with interactive visualizations, providing a lightweight and interpretable method to detect and review potential attacks in IoT environments.