Project Report
Intrusion Detection System

1. Project overview

This project implements a network Intrusion Detection System (IDS) that monitors traffic, detects suspicious activities, and presents alerts through a web-based dashboard. It combines Suricata's packet inspection engine with a Python Flask application to give administrators an easier way to view, analyze, and manage security events in real time.

The work was completed as part of the BSc Cyber Forensics program at the School of Technology and Applied Sciences, Pathanamthitta, under Mahatma Gandhi University.

2. Background and motivation

Modern organizations depend heavily on networked systems and online services, which exposes them to threats like malware, intrusion attempts, data theft, and denial-of-service attacks. Traditional security tools such as firewalls are not always sufficient to detect complex or internal attacks, so additional monitoring and detection layers are required.

Intrusion Detection Systems address this need by continuously analyzing network traffic and system behaviour to find signs of malicious or abnormal activity. Combining signature-based and anomaly-based techniques helps detect both known and new attack patterns.

3. Problem statement and objectives

Problem

Many small and medium environments lack a simple, affordable IDS with a clear dashboard for viewing alerts and understanding what is happening on the network. Security teams may struggle to interpret raw logs generated by tools like Suricata without a structured interface or database.

Objectives

The main objectives of this project were:

- To design and implement an IDS using Suricata for network traffic inspection.
- To build a Flask-based web interface to visualize alerts, logs, and basic statistics.
- To store Suricata alerts in a structured database for easier querying and analysis.
- To allow basic management of detection rules and system settings through the application.

4. Tools, technologies, and methodology

Technologies used

- Suricata – open-source IDS/IPS/NSM engine for real-time packet capture and rule-based detection.
- Python Flask – lightweight web framework used to develop the dashboard and API endpoints.
- Database – tables designed for alerts, rules, user logs, and configuration settings (relational structure).

Methodology

The project followed these high-level steps:

- Studied IDS concepts, Suricata features, and Flask basics.
- Designed the system architecture combining Suricata, a log-processing layer, and the web UI.
- Defined database tables for alerts, detection rules, user actions, and system configuration.
- Implemented logic to capture or read Suricata alert outputs and insert them into the database.
- Built Flask routes and templates to list alerts, show details, and provide simple filtering or viewing options.

5. System design

Architecture

The system consists of the following core components:

- Network traffic monitoring – Suricata listens to network interfaces, inspects packets, and applies rule sets.

- Rule-based detection – predefined and custom rules are used to identify suspicious patterns and generate alerts.
- Alert storage – alerts are stored with fields such as timestamp, source IP, destination IP, protocol, message, and severity.
- Web dashboard (Flask) – provides pages to list alerts, view details, and potentially manage rules or settings.

Database design

Key tables include:

- Alerts – stores each detected intrusion with timestamp, IPs, protocol, message, and severity level.
- Rules – stores information about detection rules, patterns, status, and last update time.
- User logs – records administrator actions, such as logging in or modifying rules, for auditing.
- Settings – stores configurable parameters like thresholds or preferences.

6. Implementation

The implementation phase focused on connecting Suricata's output to the application and presenting data in a user-friendly format. Suricata was configured to generate JSON-style logs, which can be parsed and inserted into the alerts table.

The Flask app is responsible for:

- Handling HTTP requests and routing to different pages (home, alerts, details, etc.).
- Querying the database to retrieve alerts and related data.
- Rendering HTML templates to display lists, tables, and basic charts or summaries.

Challenges included managing log formats, ensuring consistent database schema, and keeping the interface responsive while handling real-time alerts.

7. Results and observations

The final system successfully demonstrates real-time intrusion monitoring with a usable web interface. Suricata is able to detect events based on signature rules, and these alerts are visible in the dashboard with contextual information such as IPs, protocol, and time.

The project shows that open-source tools like Suricata and Flask can be combined to create a practical IDS solution suitable for learning, lab environments, or small networks.

8. Conclusion and future work

The project achieved its goal of designing and implementing a basic Intrusion Detection System with a web-based alert management interface. It reinforces key concepts in cyber forensics, network security monitoring, and secure system design.

Possible future enhancements include:

- Adding user authentication and role-based access for the dashboard.
- Integrating anomaly-based or machine learning-based detection to complement signature rules.
- Providing more advanced filtering, reporting, and visualization options for alerts.
- Integrating with external SIEM tools to support larger environments and centralized monitoring.