Intelligent Network Security Monitor with AI Assistant

November 2025

Abstract

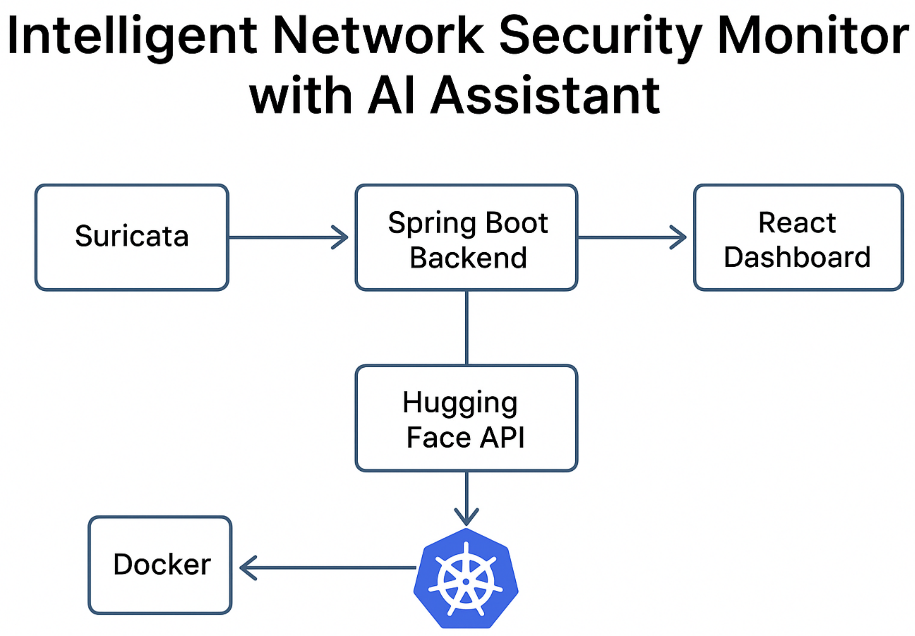
This project aims to develop a beta version of an intelligent network monitoring platform that combines real-time intrusion detection, network scanning, and AI-assisted recommendations. Using Suricata IDS, Spring Boot, React, Docker, and K3s, the system integrates the Hugging Face API to analyze security alerts and suggest corrective actions. The project will be developed collaboratively by four students over two months, focusing on essential features and practical deployment.

1 Introduction

Network monitoring and cybersecurity remain major challenges in IT infrastructures. This project proposes a lightweight, intelligent system that automatically detects network anomalies and provides recommendations using an AI assistant. The application integrates:

- **Suricata IDS** for real-time intrusion detection.
- **Spring Boot** backend for data processing and API management.
- **Hugging Face API** for generating AI-based security advice.
- **React.js Dashboard** for visualization.
- **Docker + K3s** for containerized, scalable deployment.

2 System Architecture



Component Description

1. **Suricata:** Captures network traffic and produces alerts.
2. **Elasticsearch:** Stores Suricata alerts for quick search and analysis.
3. **Spring Boot Backend:** Exposes REST APIs, runs Nmap scans, queries Hugging Face API, and communicates with Elasticsearch.
4. **React Dashboard:** Displays alerts, network map, and AI suggestions.
5. **Hugging Face API:** Interprets alerts and provides remediation text using a model like *Mistral-7B-Instruct*.
6. **Docker + K3s:** Each component runs in a container, and K3s orchestrates the microservices in a lightweight Kubernetes environment.

3 Technologies Used

- **Backend:** Java, Spring Boot, RESTful APIs.
- **Frontend:** React.js, Chakra UI.
- **AI Integration:** Hugging Face Inference API.
- **Network Tools:** Suricata IDS, Nmap.
- **Database & Indexing:** PostgreSQL, Elasticsearch.

- **Deployment:** Docker, K3s (Lightweight Kubernetes), Docker Compose.
- **Version Control:** Git + GitHub.

4 Main Features

- **Network Scanning:** Discover devices and open ports via Nmap.
- **Intrusion Detection:** Real-time detection of suspicious activities using Suricata.
- **AI Assistance:** Generate recommended countermeasures using Hugging Face models.
- **Web Dashboard:** Monitor alerts, view device information, and display AI-generated insights.
- **Containerized Deployment:** Use Docker containers managed through K3s for modular scalability.

5 Collaborative Task Distribution

Each student participates in all key areas of development through coordinated work phases:

Phase 1: Setup & Environment	All members configure Docker, K3s cluster, GitHub repo, and basic project structure.
Phase 2: Backend Development	Each member implements a module in Spring Boot: • Member 1: API structure and alert management. • Member 2: Nmap integration service. • Member 3: Hugging Face API service. • Member 4: Database + Elasticsearch integration.
Phase 3: Frontend Development	All members contribute React components: • Member 1: Dashboard layout. • Member 2: Alert visualization. • Member 3: Device list + scan results. • Member 4: AI recommendation interface.
Phase 4: Deployment & Integration	All work on Docker images, Compose setup, and deploy microservices on K3s cluster. Testing and debugging are shared.
Phase 5: Documentation & Presentation	Each contributes to the technical documentation, architecture diagram, and final project demo.

6 Project Planning (2-Month Schedule)

Week	Objective	Team Work
1	Define architecture, prepare Docker & K3s setup, initialize Git repository.	All
2	Install Suricata, set up Elasticsearch, test alert generation.	All
3	Implement Spring Boot base (REST APIs, database connection).	All
4	Integrate Nmap scanning & Hugging Face API into backend.	All
5	Build frontend components (alerts, dashboard, AI panel).	All
6	Connect frontend with backend endpoints.	All
7	Deploy using Docker Compose and K3s, perform full integration testing.	All
8	Write final documentation, test demo, prepare presentation.	All

7 Expected Deliverables

- A functional beta version of the intelligent network security platform.
- Working integration between Suricata, Spring Boot, Hugging Face, and React.
- Dockerized and K3s-deployed environment.
- Technical documentation and demonstration.

8 Future Enhancements

- Role-based authentication and user management.
- Advanced anomaly detection models.
- Integration with SIEM tools.
- Deployment in hybrid cloud environments.

9 Conclusion

This collaborative two-month project demonstrates the integration of cybersecurity tools, AI, and containerized orchestration in an educational context. By combining Docker and K3s, the system achieves modular scalability and maintainability. The inclusion of the Hugging Face API enhances automation in cybersecurity by enabling intelligent, data-driven responses to network threats.