Acknowledgement

We are proud to have completed this project and would like to express our sincere gratitude to Medhavi College and Pokhara University for providing us with the opportunity to showcase our potential and creativity through this project. We are especially thankful to our supervisor, **Mr. Sandip Oli**, for his continuous guidance, encouragement, and support throughout the project development process.

This project has contributed significantly to our academic and practical growth, challenging us to step beyond our comfort zones and enhance our technical and problem-solving skills. We would also like to acknowledge our teammates for their collaboration, dedication, and teamwork, which played a crucial role in the successful completion of this project.

Introduction

In today's cybersecurity landscape, web applications remain one of the most frequently targeted attack surfaces, making effective log monitoring and threat detection essential for organizations of all sizes. This project aims to design and develop a lightweight, web-focused Security Information and Event Management (SIEM) system capable of collecting, normalizing, analyzing, and visualizing security logs in real time. By combining a Rust-based ingestion server, Python analytics engine, and a React-powered dashboard, the system provides centralized monitoring, rule-based alerting, and actionable insights for detecting common web attacks. The primary objective is to create an accessible, extensible, and beginner-friendly SIEM tool that helps developers and security learners better understand application-level threats and defensive monitoring.

Traditional SIEM solutions are often complex, resource-intensive, and expensive, making them unsuitable for small organizations, academic environments, or learners who want to understand how security monitoring systems work internally. This gap highlights the needfor a lightweight, customizable, and educational SIEM-like platform that focuses on core concepts such as log ingestion, normalization, rule-based detection, and visualization, without the overhead of enterprise-grade systems.

This project aims to design and develop a simplified, web-focused SIEM system that demonstrates essential security monitoring principles. The system integrates a Rust-based log ingestion server, a Python analytics engine, and a React-based dashboard to provide real-time log analysis, alerting, and visualization. By emphasizing clarity, modularity, and ease of use, the project serves both as a functional security monitoring tool and a learning platform for students and developers interested in cybersecurity, backend systems, and defensive monitoring.

Objectives

- To design and develop a lightweight SIEM-inspired security monitoring system capable of collecting, normalizing, and analyzing system logs from multiple sources through a centralized backend.
- To implement a user-friendly React-based dashboard that provides real-time alerts, visual analytics, and searchable log data to support efficient threat detection and incident review.

Scope

- Collect and normalize log data from multiple sources (applications, servers, and network devices).
- Detect suspicious activities using predefined correlation rules and basic anomaly checks.
- Provide a centralized dashboard for visualizing alerts, event trends, and system status.
- Enable basic user authentication and role-based access to the monitoring dashboard.
- Support integration with external tools or APIs for future expansion.

Tools & Technologies

- **Rust**: Used for building the high-performance log ingestion server responsible for receiving, parsing, and forwarding logs.
- **Python**: Utilized for the analytics engine, including log normalization, rule-based detection, and alert generation.
- **React.js**: Framework for building the interactive web dashboard used to visualize logs, alerts, and real-time data.
- **Node.js**: For building auxiliary services or testing log-sending scripts during development.
- **MongoDB**: Used as the primary storage backend for logs, events, and alert data.
- **WebSockets**: Enables real-time updates between the backend and the dashboard.
- **REST API**: Facilitates communication between frontend and backend for fetching log data, alerts, and search results.
- **Git & Github**: Version control and collaboration platform for managing project development.
- **Linux & Windows**: Required for development, deployment, and testing of backend components for cross-functionality
- **Docker**: For containerized deployment and easier environment setup.

Problem Statement

Modern organizations generate massive volumes of logs from applications, servers, network devices, and security tools. However, small teams, academic environments, and independent developers often lack access to affordable and understandable platforms to effectively ingest, analyze, and visualize this data. As a result, security incidents, anomalous behavior, and application-level threats may go unnoticed or be detected too late.

Although enterprise-grade SIEM solutions exist, they are often complex, resource-intensive, and difficult to customize for learning or small-scale use. This project addresses this gap by developing a lightweight, modular SIEM-like system that focuses on core security monitoring capabilities such as centralized log collection, rule-based detection, and real-time visualization through a web-based dashboard.

System Architecture Design

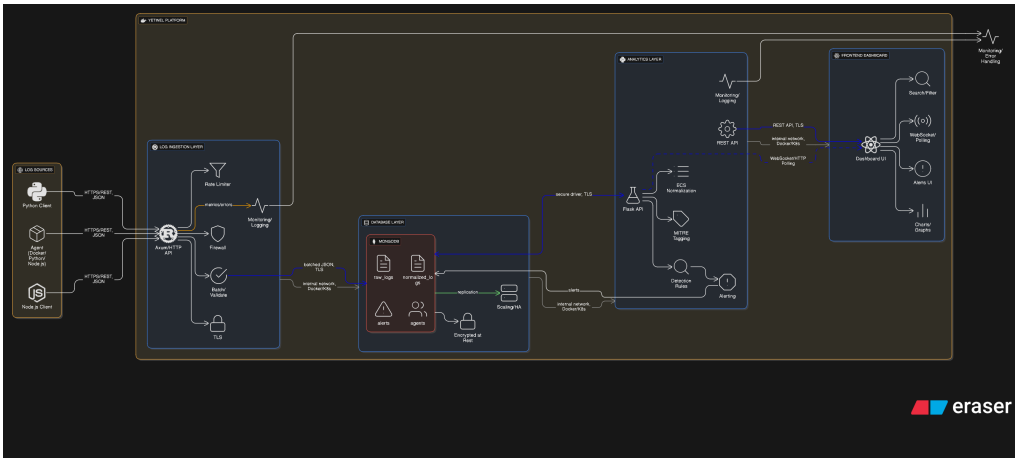

Figure 1: System Architecture of the Proposed SIEM Platform