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.

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 and academic environments often lack an accessible, lightweight, and customizable platform to ingest, analyze, and visualize these logs in a meaningful way. Without centralized log collection and real-time detection capabilities, identifying security incidents, performance issues, or anomalies becomes difficult and inefficient.

This project aims to solve this limitation by developing a simplified SIEM-like system that can ingest logs, normalize them, detect suspicious patterns, and present the data through an interactive dashboard, enabling efficient monitoring and analysis.