Simple Python SIEM - Documentation

# Overview

This document explains the structure and functionality of a simple SIEM (Security Information and Event Management) system written in Python. It simulates log generation, real-time monitoring, parsing, storing, and alerting.

# Project Structure

The project folder contains the following files and directories:

simple\_siem/  
 ├── main.py # Entry point: runs generator and monitor  
 ├── collector.py # Simulates log collection  
 ├── generator.py # Generates logs in real-time (simulated)  
 ├── monitor.py # Monitors the log file for new events  
 ├── parser.py # Parses raw logs into structured format  
 ├── storage.py # Stores logs into a local SQLite database  
 ├── alert.py # Detects alert-worthy events  
 ├── logs/  
 │ └── raw\_logs.txt # File containing generated raw logs  
 └── backend/  
 └── siem.db # SQLite database storing structured logs

# main.py

This file serves as the main entry point of the SIEM. It creates two threads: one to simulate log generation and another to monitor the log file in real-time.

Key Functions:

* - Starts log generator from generator.py
* - Starts file monitor from monitor.py
* - Uses an infinite loop to keep the program running

# generator.py

This module simulates real-time log generation. It writes logs to 'logs/raw\_logs.txt' every 5 seconds, mimicking login and file access events.

# monitor.py

This module continuously monitors 'logs/raw\_logs.txt' for new lines. When a new log entry is added, it parses, stores, and checks for alerts.

# parser.py

Parses each log line into a structured format (dictionary) with three fields:

* - timestamp
* - event\_type
* - description

# storage.py

Initializes and connects to a SQLite database located at 'backend/siem.db'. Each parsed log is inserted into the 'logs' table for storage.

# alert.py

Checks logs for specific patterns and conditions. For example, it alerts if two or more 'LOGIN\_FAIL' events occur.

# How to Run the SIEM

1. Open a terminal in the project folder

2. Run the program using:

python main.py

3. Logs will be generated and monitored in real-time

# How to Stop the SIEM

To stop the SIEM, press Ctrl+C in the terminal. This will interrupt the infinite loop and gracefully exit the program.