Divyanshu Sharma
Divyanshu08.92Sharma@gmail.com

Cybersecurity Project

Log Analysis For Security Events

—

Notes

Objective:

Build a log analyzer that detects security events from system logs, providing real-time monitoring, filtering, and actionable insights for cybersecurity professionals.

Foundational Understanding

Logs & their importance in Cybersecurity

Logs are records of system, network, or application activities that provide a trail of events happening within a system.
They are crucial in cybersecurity because they help with

- Threat Detection & Investigation: Identifying suspicious activities and tracing cyberattacks.
- Compliance & Auditing: Meeting security standards.
- Incident Response: Understanding security breaches and taking corrective actions.
- System Performance Monitoring: Ensuring systems run smoothly and troubleshooting issues.

Types of Logs & Examples in Major OS

Each operating system maintains different types of logs that serve critical security functions:

Windows Logs

Stored in Event Viewer (eventvwr.msc), categorized as:

- Security Logs: User logins, failed login attempts (Security.evtx).

- System Logs: System startup, shutdown, hardware failures (System.evtx).
- Application Logs: Errors from software like browsers, security tools (Application.evtx)

macOS Logs

Stored in /var/log/ and can be accessed via Console.app

- System Logs: System crashes, kernel logs (system.log).
- Application Logs: Errors and status of applications (/Library/Logs/)
- Console App: More System Logs in Console App

Linux Logs

Stored in /var/log/.

- Authentication Logs: Login attempts (/var/log/auth.log or /var/log/secure).
- System Logs: Kernel, boot, system events (/var/log/syslog or /var/log/messages).
- Application Logs: Errors from web servers, security tools (/var/log/httpd/access.log).

Development Progress: Building the Log Analyzer

Features Implemented So Far

- Efficient Log Processing: Handles large security logs without excessive memory usage.
- Keyword-Based Filtering: Detects critical security events (failed logins, errors, attack attempts).
- Real-Time Log Monitoring: Continuously analyzes logs as they are generated, like a lightweight SIEM tool.
- Multi-Format Log Exports: Supports TXT, JSON, and CSV for seamless integration with security dashboards.

How These Features Work

- Real-time monitoring captures logs as they appear in /var/log/, event viewer, or Console.app.
- Memory-efficient processing prevents excessive RAM usage by streaming logs line by line instead of loading entire files.

- Exporting logs in multiple formats allows easy integration with SIEM tools or forensic analysis software.

Next Steps in Development

🔹 ***Severity-Based Filtering*** *– Categorize logs by severity: INFO, WARNING, ERROR, CRITICAL.*
🔹 ***Automated Alerts*** *– Notify users when suspicious activity appears in logs (e.g., email or Slack notifications).*
🔹 ***Security Dashboard Integration*** *– Visualize log data for better insights.*
🔹 ***Correlation Techniques*** *– Identify attack patterns across multiple logs (e.g., failed SSH logins from the same IP).*

Ongoing Documentation & Updates

This documentation will evolve alongside the project, maintaining an up-to-date record of implemented features, enhancements, and planned future updates.