Log Integrity Monitoring System with Real-Time Detection

Submitted To:

Mr Abdullah Farooq

Submitted By:

Sumaiya Arshad – 231279

Marryum Afzaal - 233100

Date:

21 December 2025

Semester End Project

Table of Contents

Abstract

Modern operating systems generate large volumes of log data that are critical for detecting intrusions, unauthorized access, and system misuse. However, attackers often attempt to modify or delete log files to cover their tracks. This project presents an Advanced Log Integrity Monitoring System designed for Kali Linux that performs real-time file monitoring, hash-based integrity verification, and security event detection using a professional graphical interface.

The system continuously monitors system logs, user directories, configuration files, and web server directories. Any file creation, modification, or deletion is immediately detected and logged. Additionally, the system analyzes log contents to identify suspicious activities such as failed login attempts and sudo usage. A comprehensive report generation feature allows security analysts to review system activity in a structured format.

1. Introduction

With the increasing number of cyber attacks, system logs play a critical role in detecting unauthorized activities and security breaches. Attackers often attempt to modify or delete log files to hide their traces. Therefore, maintaining the **integrity of log files** is essential for system security.

This project aims to develop a **lightweight, standalone, real-time log integrity monitor** that works natively on Kali Linux, providing immediate alerts and professional reporting through an intuitive GUI.

2. Objectives

The primary objectives of this project are:

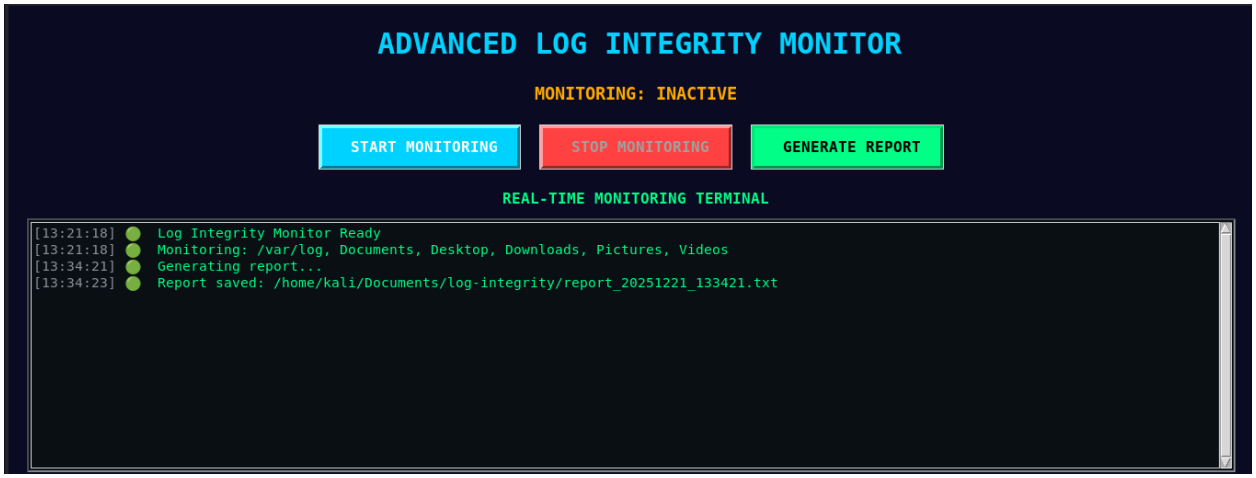
- To monitor critical system and user directories in real time
- To detect unauthorized file creation, modification, and deletion

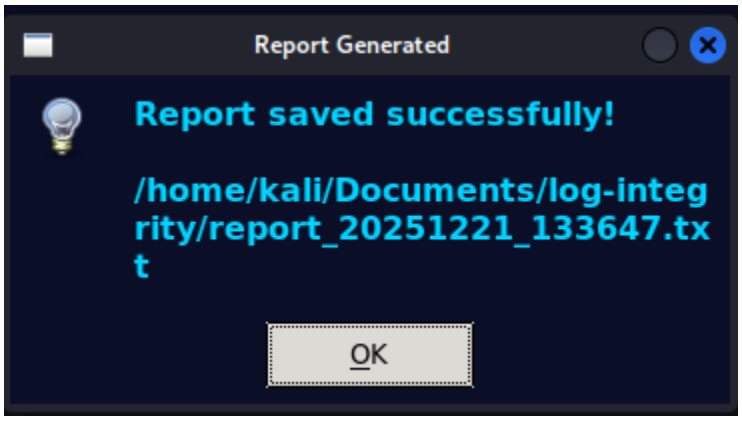
- To verify file integrity using cryptographic hash functions
- To identify security-related events from log files
- To provide a professional GUI for monitoring and visualization
- To generate detailed security reports for forensic analysis

3. System Architecture

Components Overview

Component	**Description**
GUI Module	Tkinter-based professional interface
Monitoring Engine	Real-time filesystem monitoring
Hash Engine	SHA-256 integrity verification
Database	SQLite event and hash storage
Security Analyzer	Detects suspicious log activity
Report Generator	Generates structured reports


ADVANCED LOG INTEGRITY MONITOR
MONITORING: INACTIVE
START MONITORING
STOP MONITORING
GENERATE REPORT
REAL-TIME MONITORING TERMINAL
[13:21:18] Log Integrity Monitor Ready
[13:21:18] Monitoring: /var/log, Documents, Desktop, Downloads, Pictures, Videos
[13:34:21] Generating report...
[13:34:23] Report saved: /home/kali/Documents/log-integrity/report_20251221_133421.txt


Report Generated
Report saved successfully!
/home/kali/Documents/log-integrity/report_20251221_133647.txt
OK

4. Technologies Used

Technology	Purpose
Python 3	Core programming language
Tkinter	GUI development
Watchdog	File system event monitoring

SQLite	Event and hash storage
SHA-256	File integrity verification
Kali Linux	Security-focused OS

5. Working Methodology

When the monitoring is started:

1. The system registers critical directories such as /var/log, /etc, and user folders.
2. Any file creation, modification, or deletion triggers an event.
3. For modified files, the new hash is compared with the stored hash.
4. If a mismatch is detected, an alert is generated.
5. Log files are scanned for suspicious keywords such as failed login attempts.
6. All events are stored in the database.
7. The user can generate a detailed report at any time.

6. Features Implemented

- Real-time monitoring using Watchdog
- Hash-based integrity verification
- Severity-based alerts (Low / Medium / High)
- Professional scrolling terminal interface
- Security log analysis
- SQLite-backed persistence
- One-click report generation

7. Security Use Cases

- Detecting unauthorized log deletion
- Identifying brute-force login attempts
- Monitoring sudo misuse
- Detecting tampering with system configuration files
- Web server file integrity monitoring

8. Results and Testing

Test Scenarios

- Creating a file in Documents → Detected
- Modifying /var/log/auth.log → Detected
- Deleting log file → High severity alert
- Multiple failed logins → Security alert generated

9. Limitations

- Large files are not hashed to avoid performance issues
- The system depends on OS permissions (some logs require root access)
- Keyword-based log analysis may miss advanced attack patterns
- GUI performance may reduce on very large file systems

10. Future Enhancements

- Email or Telegram alerts
- Cloud-based log storage
- Machine learning anomaly detection
- Dashboard analytics
- Role-based access control

11. Conclusion

This project demonstrates a complete and practical implementation of a log integrity monitoring system suitable for security labs and academic environments. By combining real-time monitoring, cryptographic integrity checks, and security log analysis, the system effectively enhances system visibility and forensic readiness.

12. References

- https://github.com/topics/file-integrity-monitoring

- **https://github.com/S1LV3R-C4P/File-Integrity-Checker-Tool**
- **https://github.com/catatsuy/kekkai**