Log File Analyzer for Intrusion Detection

Introduction

Log files play a crucial role in cybersecurity as they capture detailed records of system events, network activities, user authentication attempts, and application behavior. Every action performed on a system leaves a trace in log files, making them one of the primary sources for detecting security incidents. Security Operations Center (SOC) teams continuously monitor logs to identify unauthorized access, brute-force attacks, scanning attempts, malware activity, and denial-of-service attacks.

This project focuses on developing a Python-based Log File Analyzer for Intrusion Detection. The tool analyzes Apache web server logs and SSH authentication logs to identify suspicious patterns that may indicate malicious activity. The objective is to provide a beginner-friendly yet practical approach to intrusion detection using log analysis.

Abstract

The Log File Analyzer for Intrusion Detection is a Python-based security tool designed to parse and analyze Apache and SSH log files to detect potential cyber threats. The tool uses pattern matching, string analysis, and request frequency calculations to identify brute-force login attempts, repeated access requests, and suspicious HTTP response codes such as 404 and 500.

Detected security events are structured into a pandas DataFrame, exported as a CSV-based incident report, and visualized using bar graphs for better interpretation. This project demonstrates how log analysis can be used as a foundational intrusion detection technique in real-world security monitoring environments.

Tools Used

- Python – Core programming language used for log parsing and analysis
- Regular Expressions (re module) – Used for pattern matching in log entries
- Pandas – Used for organizing, analyzing, and exporting log data
- Matplotlib – Used for visualizing detected security events
- Apache and SSH sample log files – Used to simulate real-world server logs

Steps Involved in Building the Project

1. Requirement Analysis: Understood the objective of detecting suspicious activities such as brute-force attacks and DoS behavior from log files.
2. Log Collection: Collected sample Apache web server logs and SSH authentication logs to simulate realistic system activity.
3. Log Loading: Loaded the log files into the Python environment using file handling techniques.

4. SSH Log Analysis: Analyzed SSH logs to detect multiple failed login attempts, indicating possible brute-force attacks.
5. Apache Log Analysis: Parsed Apache logs to extract IP addresses, request types, and HTTP response codes.
6. Attack Detection Logic: Implemented logic to identify repeated requests from the same IP address to detect potential DoS or scanning behavior.
7. Suspicious Event Identification: Flagged abnormal HTTP response codes such as 404 and 500 errors.
8. Data Structuring: Stored all detected suspicious events in a pandas DataFrame for structured analysis.
9. Report Generation: Exported the analyzed data into a CSV-based incident report.
10. Visualization: Created bar graphs to visualize the frequency of detected attacks and anomalies.

Conclusion

The Log File Analyzer for Intrusion Detection project demonstrates how effective log analysis can support early detection of cyber attacks using simple yet powerful techniques. By leveraging Python, regular expressions, and data analysis libraries, the project successfully identifies brute-force attempts, suspicious web activity, and potential denial-of-service behavior.

This project provides a strong foundation for understanding intrusion detection concepts and Security Operations Center (SOC) workflows. It can be further enhanced by adding real-time log monitoring, integration with threat intelligence feeds, alerting mechanisms, and a web-based dashboard for improved usability.