Project Report : Log File Analyzer for Intrusion Detection

Log File Analyzer for Intrusion Detection

Name: Yash Chandrashekhar Karnik

Abstract

This project aims to design and develop a Log File Analyzer that assists in detecting potential intrusions by examining server log files. By analyzing entries from apache.log and auth.log, the system identifies abnormal access patterns, failed login attempts, and other suspicious activities that may indicate security breaches. The tool provides an efficient approach to monitoring network and server activities for early detection of attacks.

Introduction

Intrusion detection plays a critical role in ensuring the security of systems and networks. Log files, such as Apache web server logs and authentication logs, contain valuable information about user actions, IP addresses, request types, and response statuses. By analyzing these logs systematically, potential threats can be identified and mitigated before they escalate. The Log File Analyzer project focuses on extracting, parsing, and interpreting log data to detect anomalies that indicate possible intrusions.

Tools Used

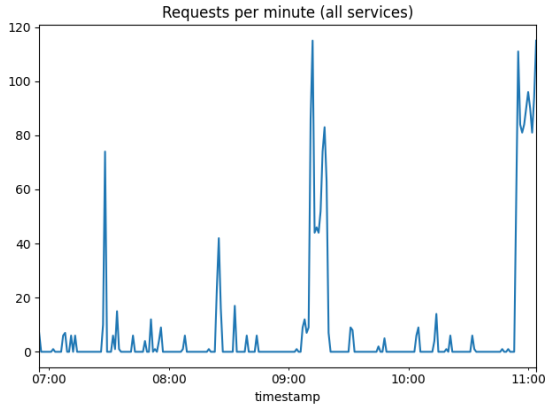
- Python 3 – For programming and log analysis
- Pandas – For reading and processing log data
- Matplotlib – For visualization and analysis of log statistics
- Kaggle Dataset – Sample Apache and authentication log data for testing
- Linux Terminal (Kali/Parrot OS) – For executing scripts and managing files

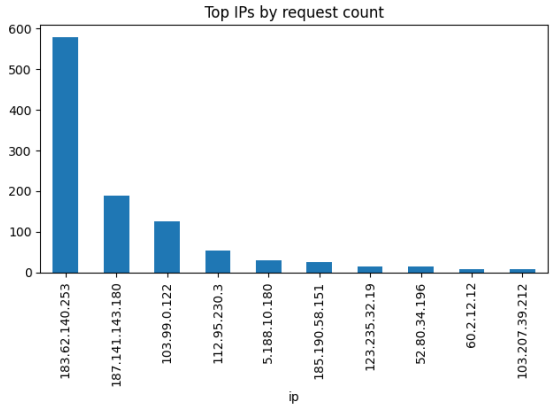
Steps Involved in Building the Project

1. Data Collection: Obtained Apache and authentication log files (apache.log, auth.log) from open-source datasets.
2. Environment Setup: Installed required dependencies (pandas, matplotlib) and configured Python virtual environment.
3. Parsing Logs: Developed a Python script to read log files line by line and extract key attributes such as timestamps, IP addresses, and request types.
4. Anomaly Detection: Identified suspicious patterns such as repeated failed login attempts or multiple requests from a single IP.
5. Visualization: Generated bar charts and summaries showing frequency of requests, top IPs, and detected alerts.
6. Result Generation: Produced structured reports (incidents.csv, alerts.log) highlighting potential intrusion indicators.

Project Report : Log File Analyzer for Intrusion Detection

Output Images





1. requests_per_minute.png

What it is

- It's a **time-series plot** (line chart) showing **number of requests per minute** (or per time interval) across the logs you supplied.
- On the **x-axis**: time (in minutes or time intervals).
- On the **y-axis**: the count of all requests (Apache + SSH, or whichever services were parsed) in each minute interval.

What it shows / Insights

- **Normal baseline**: For a typical server, most minutes will show moderate, relatively stable request volumes.
- **Spikes / bursts**: Sudden upward spikes in the request count may indicate **DoS / high-volume attacks**, crawling, or a sudden surge in traffic (legitimate or malicious).
- **Patterns / periodicity**: If you see regular oscillations (e.g., every hour), that could be scheduled jobs or bots.
- **Quiet periods**: Flat or near-zero regions indicate low or no traffic.

How to use it for intrusion detection

- Mark spikes over threshold (script's detection logic) and check the corresponding time windows for attacker IPs.
- Correlate spike windows with the incidents in your incidents.csv — if a DoS alert is flagged during a high-count minute, that matches.
- Spot patterns like repeated micro-spikes (e.g., many small surges every few minutes) — could be low-rate attacks or scanning.

2. top_ips.png

What it is

Project Report : Log File Analyzer for Intrusion Detection

- A **bar chart** showing the **top N IP addresses** sorted by their total number of requests over the entire log period (default N = 10).
- **x-axis**: IP addresses (top talkers).
- **y-axis**: request count (how many total requests each IP made).

What it shows / Insights

- **Heavy hitters**: The IPs that contribute most of the traffic. Often these include search engine bots, specific users, or potential attackers.
- **Relative volume**: The difference in height between bars shows relative intensity. If the top IP has 10× requests over second place, that is noteworthy.
- **Blacklist correlation**: If an IP on this bar chart is also flagged in incidents.csv, it strengthens suspicion.

How to use it for intrusion detection

- For each IP in the top list, inspect the detailed logs (URLs accessed, status codes, and timestamps) to determine whether their behavior is normal.
- If a top IP is also blacklisted or flagged in incident alerts, that's a red flag.
- Use this bar chart to filter and prioritize investigation — e.g. start with top IPs.

Conclusion

The Log File Analyzer project successfully demonstrates how analyzing log data can help identify and mitigate security threats in real time. By leveraging Python-based data analysis and visualization, the project enhances understanding of intrusion detection techniques and showcases the importance of log monitoring in cybersecurity operations. Future improvements could include integrating real-time monitoring and alerting mechanisms using machine learning models.

Submitted by:
Yash Chandrashekhar Karnik