Log File Analyzer for Intrusion Detection

---

1. Introduction

In today's digital landscape, servers are constantly targeted by automated scripts and malicious actors. These activities often leave traces in system log files. This project aims to build a Python-based tool to analyze Apache-style logs and detect potential security threats such as brute-force login attempts, SQL injections, cross-site scripting (XSS), and **Denial of Service (DoS)** attacks. The tool not only identifies suspicious patterns but also generates visual insights and structured incident reports for further review.

---

2. Abstract

The Log File Analyzer is designed to process and analyze web server logs for potential intrusion activities. The system uses regular expressions to parse raw logs into structured format, applies rule-based logic to flag suspicious behavior, and visualizes key metrics such as IP activity and request volume.

In addition to identifying brute-force, SQL injection, and XSS patterns, the tool detects **DoS attacks** by flagging IP addresses that generate abnormally high request volumes. The final output includes a suspicious log CSV, an incident summary, and multiple charts. This project improves manual log review efficiency and demonstrates how simple data processing can contribute to basic intrusion detection.

---

3. Tools Used

Tool / Library	Purpose
Python 3.x	Core scripting language
pandas	Data analysis and transformation
matplotlib	Visualization (plots, pie charts)
seaborn	Enhanced statistical visualizations
re (regex)	Parsing logs with regular expressions

---

4. Steps Involved in Building the Project

a. Log Parsing (parser.py)

- Reads Apache access logs line by line
- Uses regex to extract IP, timestamp, method, URL, status, and size
- Converts timestamps to datetime format
- Saves parsed logs to parsed_logs.csv

b. Suspicious Activity Detection (detection.py)

- Loads parsed log data
- Applies detection rules:
    - Status codes 401, 403 → failed logins (brute-force)
    - SQL injection patterns in URLs (', OR, SELECT, etc.)
    - XSS patterns (e.g. <script>)
    - **DoS detection based on high request volume (>30 requests/IP)**
- Flags and saves suspicious activity into suspicious_logs.csv and incident_report.csv

c. Visualization (visualise.py)

- Plots:

    - Top 10 IPs by request count
    - Attack type distribution (pie chart)
    - HTTP status code breakdown (pie chart)
    - Request volume over time (line graph)
    - **Top IPs by traffic for DoS detection (bar chart)**
- All plots are saved in reports/plots/ as .png files

---

5. Conclusion

This project showcases how basic data analysis techniques can assist in intrusion detection using log files. By automating the parsing, threat identification, and visualization of logs, the tool provides a foundational layer for monitoring server activity.

In addition to traditional attack vectors like brute-force and injection attacks, the inclusion of **DoS detection** adds another dimension of security analysis, identifying high-volume IP activity.

This system can serve as a lightweight alternative or a complementary system to larger SIEM platforms. The modular design also allows for future improvements such as SSH log analysis, threat scoring, and integration with threat intelligence feeds.
