**Project Report: Log File Analyzer for Intrusion Detection**

**Name:** Aryanshu Saha  
**Final Submission Date:** 28 July 2025  
**GitHub Repository Link:** <https://github.com/Aghorix88/Elevate-Lab-internship-Tasks.git>

**1. Introduction**

With increasing cybersecurity threats, organizations must proactively monitor system and network logs to detect and mitigate unauthorized access attempts. This project focuses on developing a log file analyzer tool for Linux systems that automates the detection of brute-force login attempts, denial-of-service (DoS) patterns, and web scanning activity through Apache and SSH logs. The analyzer provides visual insights and generates incident reports, making it ideal for SOC teams and Linux administrators.

**2. Abstract**

This Python-based tool parses and analyzes server log files — /var/log/auth.log and /var/log/apache2/access.log — to identify suspicious activities such as repeated failed SSH login attempts and excessive web server hits. Using pandas, regex, and matplotlib, the tool generates detailed threat reports and visualizations. It integrates with AbuseIPDB (a public blacklist API) for reputation scoring of flagged IP addresses and supports automated daily execution via cron scheduling.

**3. Tools & Technologies Used**

| **Tool/Library** | **Purpose** |
| --- | --- |
| Python 3 | Scripting Language |
| Kali Linux 2025.2 | Operating Environment |
| pandas | Data parsing and aggregation |
| matplotlib/seaborn | Visualization |
| regex (re) | Log pattern matching |
| requests | Public IP reputation lookup (AbuseIPDB) |
| cron | Automation of daily execution |
| GitHub | Project versioning and submission |

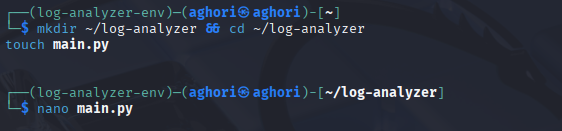
**4. Steps Involved in Building the Project**

**Step 1: Environment Setup**

* Created a virtual environment using venv
* Installed required Python packages via pip

**Step 2: Log File Preparation**

* Used /var/log/auth.log and /var/log/apache2/access.log (or fallback test logs if unavailable)

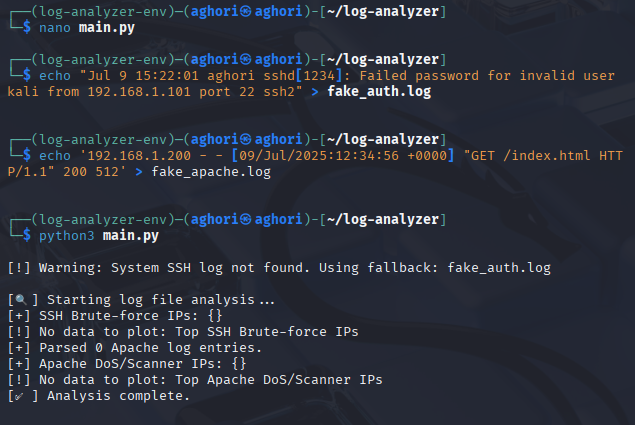


**Step 3: Parsing & Detection Logic**

* Parsed SSH logs to detect failed login attempts per IP
* Parsed Apache logs to identify DoS or scanning behavior
* Set configurable detection thresholds

**Step 4: Visualization**

* Generated bar charts showing top attacker IPs using matplotlib & seaborn



**Step 5: Reporting**

* Exported flagged IPs and attack data into CSV and JSON files

**Step 6: Blacklist Integration**

* Integrated AbuseIPDB API to check if detected IPs are listed as malicious

**Step 7: Automation with Cron**

* Configured a daily cron job at 3:00 AM to run the tool automatically

**Step 8: Version Control**

* All source code and reports uploaded to a private/public GitHub repository

**5. Conclusion**

This project successfully delivered an end-to-end intrusion detection system for Linux server logs. It automates the detection of brute-force attacks and web scans, generates reports and visualizations, and integrates real-time threat intelligence. The tool is extendable and can be used in real-time Linux environments or integrated with SIEM pipelines for advanced detection.