Project Report: Log File Analyzer for Intrusion Detection

I. Introduction

The modern security landscape requires automated, rapid analysis of volumetric log data to distinguish malicious activity from benign noise. Every action on a website every click, every page view, and every error is recorded in a massive text file called an access log. When a security incident happens, we need to read this huge, messy file very quickly to find out what happened.

The goal of this project was to build a smart system that can read thousands of log entries and automatically turn them into simple pictures and clear facts. This allows anyone, even beginners, to understand the attack, find the main bad guys, and know exactly **when** the most dangerous part of the attack happened so we can fix the problem fast.

II. Abstract

A comprehensive analysis of the access log dataset revealed a coordinated intrusion event spanning four distinct threat categories: DoS/DDoS, Scanning/Brute Force, SQL Injection, and XSS Attacks. The project identified two overwhelmingly dominant internal threat endpoints: **192.168.4.164** and **192.168.4.25**.

Here are the most important discoveries:

1. **Temporal Masking:** The attackers started with a massive digital flood (DDoS/DoS), hitting a peak of **2,908 website requests per minute** at **15:19:00**. This flood was very short, it was a **distraction** used to overload the system and hide their real, more dangerous actions.
2. **Attribution Disparity:** The two attacking computers had different jobs. Computer **192.168.4.164** was responsible for the highly specialized, dangerous attacks (like SQL Injection and XSS), while 192.168.4.25 was mostly just helping with general scanning. This means **192.168.4.164 is the primary threat.**

The resulting visualizations validate that forensic efforts must be immediately prioritized on traffic associated with 192.168.4.164 and focused strictly on the three-minute window of system instability (15:19:00 to 15:22:00).

III. Tools Used

Category	Tool/Library	Application in Project
Core Technology	Python 3.x	The programming language used to build the entire detection system.
Parsing Method	Python Regular Expressions (re)	Core methodology used to extract features (IP, Status Code, etc.) from the raw, unstructured access.log lines.
Data Organization	Pandas	The spreadsheet tool that takes the messy log information and puts it into neat columns and rows so we can count things easily.
Visualization	Matplotlib	The drawing tool used to create clear, professional graphs that show the attack timeline and compare the bad guys.
Methodology	Structured Prompt Engineering	An advanced technique used to generate, validate, and debug production-quality code for security analytics, ensuring rapid iteration and adherence to technical requirements.
Artifacts	incident_report.csv, Threat_Visualization_Validated.png	Output reports for quantitative summary and visual forensic evidence.

IV. Steps Involved in Building the Project

Our project followed five simple steps to turn raw text logs into security action plans.

Phase	Focus	Key Technical Output
Phase 1 – Data Ingestion & Collection	Data Ingestion & Collection	Utilized Python file I/O to stream-read access.log line-by-line, optimizing memory usage for large datasets.
Phase 2 – Parsing & Transformation	Parsing & Transformation	Applied robust Regular Expressions (re) to extract structured fields (IP, timestamp, status code, request) from unstructured logs.
Phase 3 – Data Modeling & Indexing	Data Modeling & Indexing	Loaded structured data into Pandas DataFrame; converted timestamps to datetime objects and set as time-series index.
Phase 4 – Intrusion Detection (Counting)	Intrusion Detection	Implemented threat-counting logic (DoS/DDoS, Brute Force, SQLi, XSS); used Pandas groupby() to aggregate by Source IP and generated incident_report.csv.
Phase 5 – Forensic Visualization & Validation	Visualization & Validation	Generated dual-plot forensic figure using Matplotlib with proper time formatting and categorical IP attribution charts.

V. Conclusion

This project successfully created a fast, repeatable system for identifying and prioritizing digital threats. Our main conclusion is clear: **192.168.4.164 is the immediate threat priority.**

Simple Action Plan:

- **Priority Block:** We must immediately block all traffic coming from the primary attacker, **192.168.4.164**, as it is responsible for the most serious specialized attacks.
- **Focus Your Search:** Any team investigating the breach should ignore most of the log file. They only need to look at the system's records for the **three-minute window between 15:19:00 and 15:22:00**. This is when the system was overloaded, and the main attack tools were most likely successful.

This project demonstrates the effective application of structured programming and sophisticated visualization techniques to rapidly synthesize security data into clear, critical intelligence, fulfilling the mandate for security preparedness and prioritized incident response.