🔐 Linux Text Processing for Log Volume & Pattern Analysis

📌 Overview

This project demonstrates how core Linux text-processing utilities (`wc`, `sort`, `uniq`, and pipelines) can be applied to **log volume analysis, frequency detection, and data prioritization**.

Rather than relying on centralized tooling, this project focuses on **host-based analysis techniques** commonly used during early-stage incident response, system troubleshooting, and SOC triage.

🎯 Objective

- Measure log volume efficiently
- Identify high-frequency indicators (e.g., client IPs)
- Aggregate text data across files
- Rank numerical data to prioritize investigation targets

🧩 Scenario

A system administrator or SOC analyst is tasked with reviewing multiple log and data files to:

- Quickly assess the **scale of activity**
- Identify **high-frequency sources** that may indicate abuse or misconfiguration
- Aggregate data across multiple files

- Rank numerical indicators to support investigation or reporting

These tasks are often performed **before** deeper forensic or SIEM-based analysis.

🛠️ Tools & Technologies

- Linux (Bash)
- `wc`
- `sort`
- `uniq`
- Standard Unix pipelines and redirection
- Structured and unstructured text files

🔍 Methodology

1️⃣ Log Volume Assessment

- Counted the number of lines in an access log
- Stored results separately to preserve original evidence

Security relevance:
Line counts provide a fast approximation of traffic volume and investigation scope.

2️⃣ Frequency Analysis of Client Identifiers

- Extracted client identifiers (e.g., IP addresses)

- Counted frequency of occurrence
- Ranked results to identify top contributors

Security relevance:
High-frequency sources may indicate:

- Legitimate heavy usage
- Misconfigured services
- Automated scanning or abuse

3️⃣ Aggregation Across Multiple Files

- Counted total words across multiple text files in a directory
- Produced a single summarized output

Security relevance:
Demonstrates scalable analysis across file collections, useful for documentation review, data audits, or large log sets.

4️⃣ Numerical Ranking & Prioritization

- Sorted numerical values in descending order
- Extracted top results for focused review

Security relevance:
Ranking allows analysts to quickly prioritize:

- Largest values
- Highest utilization

- Most impactful indicators

🧠 Key Findings

- Line counts provide fast situational awareness
- Frequency analysis highlights dominant contributors in log data
- Aggregation simplifies analysis across multiple files
- Numerical ranking supports efficient prioritization