Log File Analysis with Bash Script

Mahmoud Amr Zaghloula 2205055

MAY 10, 2025

TASK 3

Information Security Management

Log File Analysis Report

Introduction

This report presents the analysis of an Apache web server log file (apache_logs) using a custom Bash script. The script processes the log to extract key statistics about requests, IP addresses, failures, and trends, providing insights into server performance and potential issues. The objective is to understand request patterns, identify anomalies, and suggest improvements to enhance system reliability and security.

Statistics

The following statistics were extracted from the log file, covering a total of 10,000 requests:

1. **Request Counts**:
   - Total Requests: 10,000
   - GET Requests: 9,952 (99.52%)
   - POST Requests: 5 (0.05%)
2. **Unique IP Addresses**:
   - Total Unique IPs:1753
3. **Failed Requests**:
   - Failed Requests (4xx/5xx): 220
   - Failure Rate: 2.00%
4. **Most Active User**:
   - Most Active IP: 66.249.73.135 with 482 requests (4.82% of total).
5. **Daily Request Average**:
   - Average Requests per Day: 2,500.
6. **Failure Analysis by Day**:
   - May 19, 2015: 66 failures
   - May 18, 2015: 66 failures
   - May 20, 2015: 58 failures
7. **Requests by Hour**:
   - Highest: Hour 14 (498 requests)
   - Lowest: Hour 08 (345 requests)
   - Fairly even distribution, with a peak in the afternoon (12:00–20:00).
8. **Request Trends**:
   - Peak Hour: 14:00 with 498 requests
   - Trend: Increased activity in the afternoon.

9. **Status Code Breakdown**:
    - 200 (OK): 9,126
    - 304 (Not Modified): 445
    - 404 (Not Found): 213
    - 301 (Redirect): 164
    - 206 (Partial Content): 45
    - 500 (Server Error): 3
    - 416 (Range Not Satisfiable): 2
    - 403 (Forbidden): 2
10. **Most Active IP by Method**:
    - GET: 66.249.73.135 with 482 requests
    - POST: 78.173.140.106 with 3 requests
11. **Failure Patterns**:
    - Top Failure Hours:
        - 09:00: 18 failures
        - 05:00: 15 failures
        - 06:00: 14 failures

Analysis

The log analysis reveals several key insights:

- **Request Distribution**: The overwhelming majority of requests (99.52%) are GET, indicating that users are primarily accessing static content (e.g., web pages, images). The extremely low number of POST requests (5) suggests minimal form submissions or interactive actions, which could be typical for a content-heavy site.

- **Most Active IP**: The IP 66.249.73.135 (likely a crawler, possibly Googlebot, based on its activity) accounts for 482 GET requests. This is significant but not alarming, as crawlers often generate high request volumes.

- **Failed Requests**: A 2% failure rate (220 requests) is relatively low but warrants attention. Most failures are 404 errors (213), suggesting broken links or missing resources. The 3 instances of 500 errors indicate rare server-side issues.

- **Daily Patterns**: The log covers approximately 4 days (based on the 2,500 daily average). Failure rates are highest on May 18 and 19, 2015 (66 failures each), possibly due to server maintenance, misconfigurations, or external factors like increased traffic.

- **Hourly Trends**: Requests peak at 14:00 (498 requests) and are lowest at 08:00 (345 requests). The afternoon (12:00–20:00) sees higher activity, likely corresponding to user time zones or work hours.

- **Failure Patterns**: Failures are concentrated in early morning hours (05:00–09:00), which could align with automated scripts, maintenance windows, or low server resource availability.

Suggestions

Based on the analysis, the following recommendations can improve system performance and security:

1. **Reduce Failures**:
   - **404 Errors**: Audit the website for broken links using tools like wget --spider or a link checker. Ensure all requested resources (e.g., images, pages) are available or properly redirected.
   - **500 Errors**: Review server error logs (e.g., /var/log/apache2/error.log) to identify the cause of the 3 server errors. Check for issues in server-side scripts (e.g., PHP, Python) or database connectivity.
2. **Monitor High-Failure Days**:
   - Investigate server logs for May 18 and 19, 2015, to determine if failures were due to updates, misconfigurations, or attacks. Increase monitoring during similar events in the future.
3. **Optimize for Peak Hours**:
   - Allocate additional server resources (e.g., CPU, RAM) during afternoon hours (12:00–20:00), especially around 14:00, to handle peak traffic.
4. **Security Considerations**:
   - The high activity from 66.249.73.135 is likely a legitimate crawler but should be monitored. If it’s overloading the server, consider rate-limiting crawlers using robots.txt or server rules.
   - The low number of POST requests (5) reduces the risk of brute-force attacks, but ensure forms have CSRF protection and CAPTCHA to prevent future abuse.

5. **System Improvements**:
   - Implement caching (e.g., Varnish or Cloudflare) to reduce server load for frequent GET requests, especially from crawlers.
   - Use a Web Application Firewall (WAF) to detect and block malicious traffic, particularly during early morning hours when failures peak.
   - Set up real-time monitoring with tools like Prometheus or Nagios to catch 500 errors and high failure rates immediately.

Conclusion

The analysis of the apache_logs file provides valuable insights into server performance, user behavior, and potential issues. With a low failure rate (2%) and consistent request patterns, the server appears stable but could benefit from targeted improvements. Addressing 404 errors, investigating server errors, and optimizing for peak hours will enhance user experience and system reliability. Regular monitoring and security measures will further protect the server from anomalies and ensure robust performance.