Assignment

Yumna Medhat Anter 2205231

1. Introduction

This report presents the analysis of a web server log file (access.log) using a Bash script, as per the assignment requirements. The objective is to extract meaningful statistics, identify patterns, and provide actionable suggestions for system improvement. The analysis focuses on request counts, unique IP addresses, failure rates, daily averages, and trends, while ensuring information security and data integrity. The results are derived from the log file access.log and stored in analysis_results.txt.

2. Analysis Results

The Bash script processed the log file and generated the following statistics:

2.1 Request Counts

- Total Requests: 10000
- GET Requests: 9952
- POST Requests: 5

2.2 Unique IP Addresses

- Total Unique IPs: 1753
- IP Activity (Top 5 for brevity):
    - 66.249.73.135: 482 GET, 0 POST
    - 130.237.218.86: 357 GET, 0 POST
    - 100.43.83.137: 84 GET, 0 POST
    - 108.171.116.194: 65 GET, 0 POST
    - 14.160.65.22: 50 GET, 0 POST

2.3 Failure Requests

- Failed Requests (status codes 4xx and 5xx): 220
- Failure Percentage: 2.20%

2.4 Most Active IP

Top IP: 66.249.73.135 (482 requests, 4.82% of total).

2.5 Daily Request Averages

- Number of Days: 4
- Average Requests per Day: 2500.00

2.6 Days with Highest Failures

Failure Distribution:

- 19/May/2015: 66 failures
- 18/May/2015: 66 failures
- 20/May/2015: 58 failures
- 17/May/2015: 30 failures

2.7 Requests by Hour

Hour	Request Count
14:00	498
15:00	496
19:00	493
20:00	486
17:00	484
18:00	478
13:00	475
16:00	473
12:00	462
11:00	459

Hour	Request Count
21:00	453
10:00	443
05:00	371
06:00	366
02:00	365
09:00	364
00:00	361
01:00	360
07:00	357
23:00	356
04:00	355
03:00	354
22:00	346
08:00	345

2.8 Status Code Breakdown

Status Code	Occurrences
200	9126
304	445

Status Code	Occurrences
404	213
301	164
206	45
500	3
416	2
403	2

2.9 Most Active IP by Method

- Top GET IP: 66.249.73.135 (482 GET requests)
- Top POST IP: 78.173.140.106 (3 POST requests)

2.10 Failure Patterns

Failure Times (Top 5 for brevity):

- 20/May/2015, 09:05:45: 2 failures
- 20/May/2015, 09:05:37: 2 failures
- 20/May/2015, 09:05:25: 2 failures
- 20/May/2015, 09:05:20: 2 failures
- 20/May/2015, 09:05:04: 2 failures

3. Analysis and Trends

The analysis reveals several insights:

- **Request Patterns**: The majority of requests are concentrated between 14:00 and 20:00, with a peak of 498 requests at 14:00. This could indicate high user activity or automated processes during these hours.
- **Failure Rate**: A 2.20% failure rate is moderate, with 213 instances of 404 (resource not found) and 3 instances of 500 (server error). This suggests issues with resource availability or minor server configuration problems.
- **IP Activity**: The IP 66.249.73.135 is the most active, contributing 4.82% of total requests, primarily GET requests. This could indicate legitimate usage (e.g., a search engine crawler) or require further investigation for potential abuse.
- **Data Integrity**: The log file appears consistent, with no evidence of tampering or malformed entries. The large volume of requests (10000) provides a robust dataset for analysis.

4. Suggestions for Improvement

Based on the analysis, the following recommendations are proposed:

4.1 Reducing Failures

- **Address 404 Errors**: Ensure all requested resources are available or redirect users to valid pages. The 213 instances of 404 errors suggest missing or misconfigured resources.
- **Fix 500 Errors**: Investigate server logs for the cause of the three internal server errors (e.g., on 20/May/2015). This could involve checking application code or server configuration.
- **Implement Monitoring**: Use tools like Nagios or Prometheus to monitor server health and detect errors in real-time.

4.2 Handling Peak Times

- **Resource Allocation**: Increase server resources (e.g., CPU, memory) during peak hours (14:00–20:00) to handle higher request volumes.
- **Load Balancing**: Deploy a load balancer to distribute traffic across multiple servers to manage the high volume of requests during peak times.

4.3 Information Security

- **Investigate IP 66.249.73.135**: Monitor this IP for suspicious activity, given its high contribution to requests (482 GET requests). Implement rate-limiting or IP banning if it exhibits attack patterns. Note that this IP may belong to a legitimate crawler (e.g., Googlebot).
- **Enable WAF**: Deploy a Web Application Firewall to filter malicious requests and protect against common attacks like SQL injection or DDoS.
- **Log Auditing**: Regularly audit logs to detect anomalies, ensuring data integrity by verifying log entries against expected formats.

4.4 System Improvements

- **Caching**: Implement caching (e.g., using Redis or Varnish) for frequently requested resources to reduce server load, especially during peak hours.
- **Error Handling**: Improve application error handling to provide user-friendly messages for 404 errors and log detailed diagnostics for 500 errors.
- **Log Rotation**: Configure log rotation to manage the large log file sizes, ensuring efficient storage and analysis.

5. Conclusion

The Bash script successfully analyzed the access.log file, providing valuable insights into request patterns, failure rates, and potential security concerns. The moderate failure rate and high activity from a single IP highlight areas for optimization. By implementing the proposed improvements, the system can achieve better reliability, performance, and security. Future analyses with similar large datasets could reveal more detailed trends and patterns.