Nadeen Ehab Fikry **2205081**

Log File Analysis Report

1. Request Counts

Total requests: 10000

GET requests: 9952

POST requests: 5

2. Unique IP Addresses

Number of unique IPs: 1753

GET/POST breakdown per IP (Top 5):

83.149.9.216 - GET: 23, POST: 0

24.236.252.67 - GET: 1, POST: 0

93.114.45.13 - GET: 6, POST: 0

66.249.73.135 - GET: 482, POST: 0

50.16.19.13 - GET: 113, POST: 0

3. Failure Requests

Failed requests (4xx/5xx): 220

Failure percentage: 2.20%

4. Top User

Most active IP address: 66.249.73.135 (482 requests)

5. Daily Request Averages

Average requests per day: 2500.00

6. Failure Analysis

Days with most failures:

18/May/2015 - 66 failures

19/May/2015 - 66 failures

20/May/2015 - 58 failures

Additional Insights

Requests by Hour (Top 5):

14:00 - 498 requests

15:00 - 496 requests

19:00 - 493 requests

20:00 - 486 requests

17:00 - 484 requests

Request Trends

Requests were consistently logged between 00:00 and 23:00. The peak request hour was 14:00 with 498 requests, and the lowest was 08:00 with 345 requests. On average, each hour received about 416.7 requests.

Status Code Breakdown

200: 9126 times

206: 45 times

301: 164 times

304: 445 times

403: 2 times

404: 213 times

416: 2 times

500: 3 times

Most Active User by Method

Most GET requests from: 66.249.73.135

Most POST requests from: 78.173.140.106

Failure Patterns

Failures peak at hour: 09:00

Suggestions

- Look into peak failure hours/days to identify server issues.

- Analyze request types by IP to optimize endpoint usage.

- Improve error handling or rate limiting if failures are due to overload.

- Limit excessive requests from the same IP through rate-limiting.

- Investigate server issues during peak failure times (e.g., specific hours).

- Optimize performance around traffic peaks to ensure smooth response.

- Consider bot detection for traffic surges from known crawler user agents.

- The highest failure rate occurred at 09:00 with 18 failed requests. Focus monitoring and debugging efforts around this time.

- Limit excessive requests from single IPs through rate-limiting or firewall rules.

- Optimize backend performance during peak hours to prevent service degradation.

- Analyze user-agent patterns to detect and mitigate aggressive bots or crawlers.