Information Security Management

Log file analysis

Name: Amina Ahmed Ferra

ID: 2205225

Log Source: Apache Access Log
Toolset: Kali Linux, Bash, Python

1-Summary

This task involved analysing a web server log file using a Bash script to extract key metrics and insights related to server request patterns, errors, and potential security issues. The goal was to generate actionable statistics and recommend improvements based on the data.

2-About the Log File

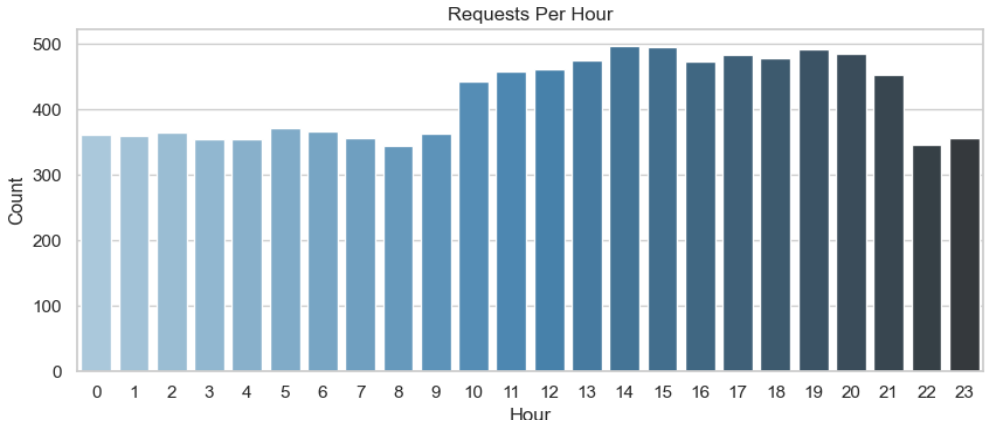
The log file analysed is an Apache access log named info, simulating real-world HTTP request records to a web server. Each line contains:

- Client IP address
- Request timestamp
- HTTP method (GET/POST)
- Requested resource
- Response status code (e.g., 200 OK, 404 Not Found)
- User agent info
- Line Count: 10,000 lines
- Time Range: Covers at least 4 days based on date analysis (17/May/2015 to 20/May/2015)

This format is widely used to track server activity for monitoring, security audits, and performance evaluation.

3-Log Analysis Results

3.1 Summary Metrics



Metric	Value
Total Requests	10,000
GET Requests	9,952
POST Requests	5
Unique Ips	1,753
Failed Requests	220 (2.20%)
Most Active IP	66.249.73.135 (482 requests)
Average Daily Requests	2,500.00

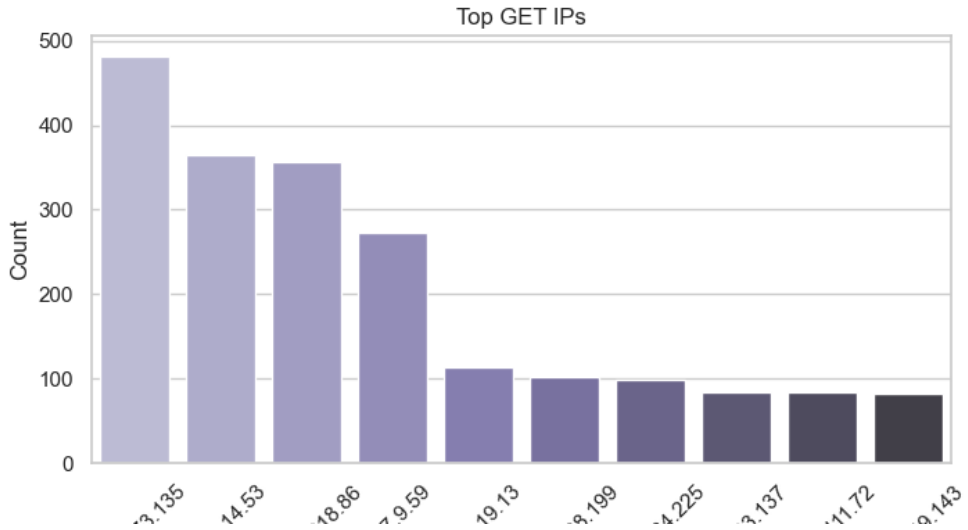
- **Observation:**
  The server handles mostly GET traffic. A single IP is particularly active, which may be a bot or crawler.
- **Security Concern:**
  Rate limiting may be needed for active users like 66.249.73.135.

3.2 Top Failure Hours

Hour	Count
18	09
15	05
14	06
12	17
12	13

- **Observation:**
  Failures are concentrated during business hours.
- **Security Concern:**
  Check if these correspond to real user load or bot activity.

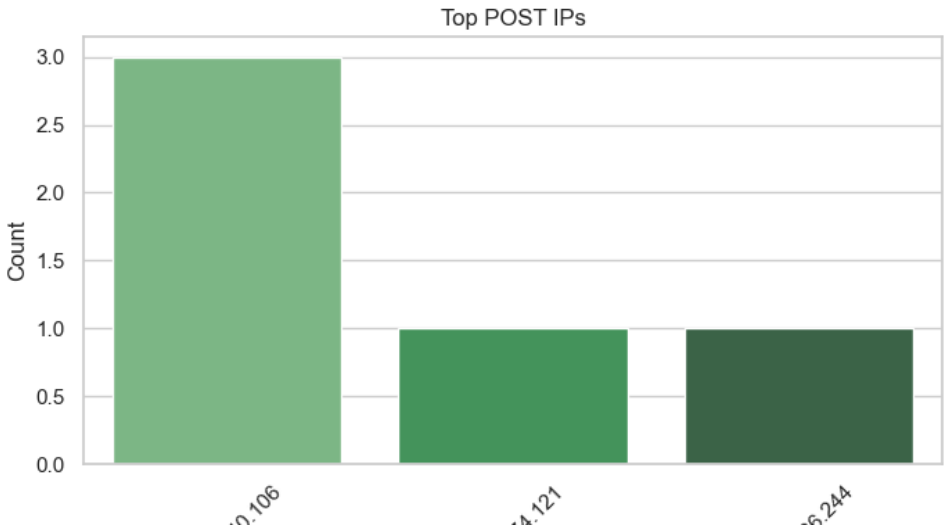
3.3 Top GET Ips



IP	GET Count
66.249.73.135	482
46.105.14.53	364
130.237.218.86	357

- **Observation:**
  Heavy GET use may indicate indexing bots or scraping.
- **Security Concern:**
  Set thresholds to block or challenge excessive traffic from single IPs.

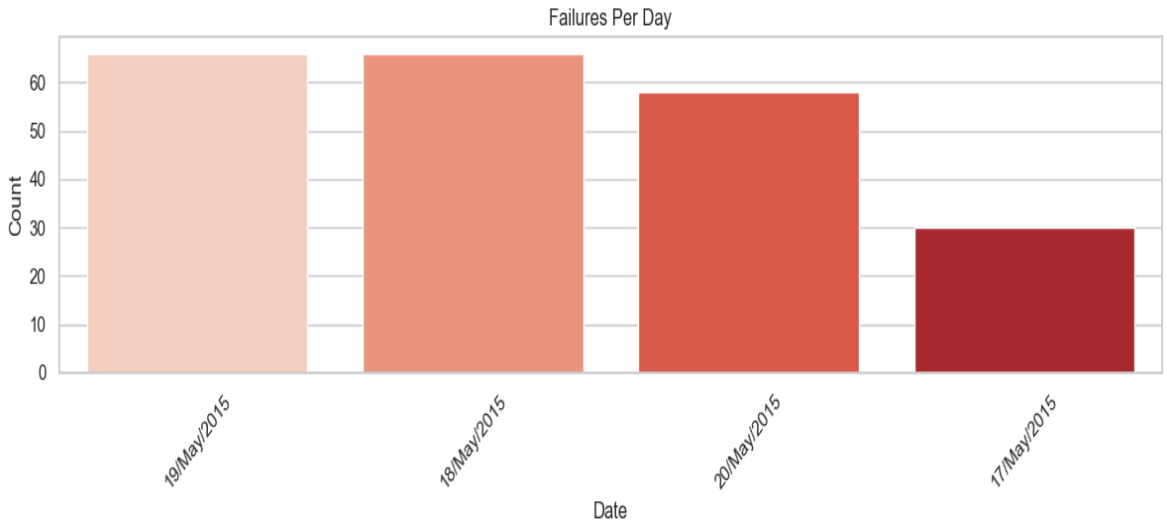
3.4 Top POST Ips



IP	POST Count
78.173.140.106	3
91.236.74.121	1
37.115.186.244	1

- **Observation:**
  Few POST requests indicate limited write operations.
- **Security Concern:**
  POSTs can be sensitive — validate sources and enforce strict endpoint security.

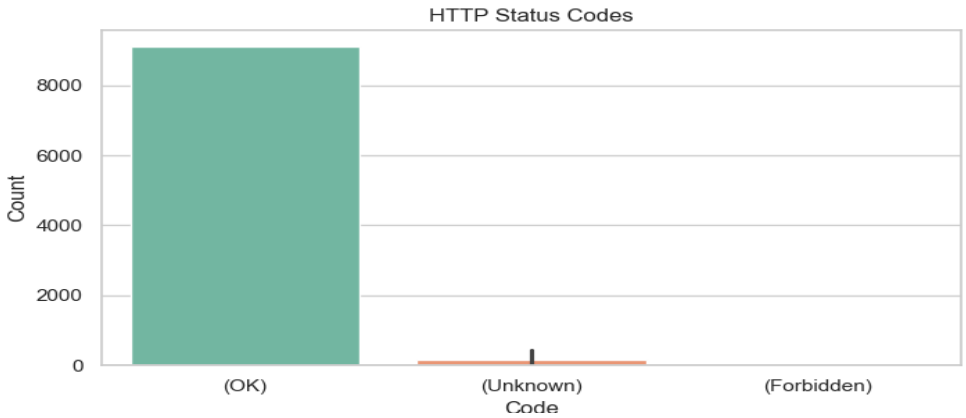
3.5 Top Failure Days



Fail Count	Date
66	19/May/2015
66	18/May/2015
58	20/May/2015
30	17/May/2015

- **Observation:**
  High failures across consecutive days suggest system instability or abuse.
- **Security Concern:**
  Investigate changes or attacks during this period.

3.6 Status Code Breakdown



Code	Meaning	Count	%
200	OK	9,126	91.26%
206	Partial	45	0.45%
301	Moved Permanently	164	1.64%
304	Not Modified	445	4.45%
403	Forbidden	2	0.02%
404	Not Found	213	2.13%
416	Range Not Satisfiable	2	0.02%
500	Internal Server Error	3	0.03%

- **Observation:**
  The presence of 404s and a few 500s may indicate misconfigurations or failed attempts to access invalid resources.
- **Security Concern:**
  Frequent 404s could be scans for vulnerable endpoints.

Recommendations

1-Reduce Failures

- Improve routing and input validation to reduce 4xx/5xx errors.
- Use error monitoring to alert on sudden error spikes.

2- Monitor Busy Periods

- Focus on 12:00–18:00 timeframe.
- Set alerts or throttle requests during those hours if needed.

3-Improve Security

- Use Web Application Firewalls (WAF) and rate limiting.
- Block or challenge high-volume requesters if they're not legitimate users.
- Enable logging for POST request payloads (where legal and secure) to investigate intent.

4-System Enhancements

- Review load balancing, caching, and scaling mechanisms.
- Add diagnostics/log enrichment for better issue tracing.

Conclusion

The analysis of the log file provided valuable insights into request patterns, failure rates, and potential security concerns. While most requests were successful, a small percentage of failures (2.20%) were identified, with certain hours and days showing higher failure rates. Additionally, a few IP addresses exhibited unusual activity, potentially indicating abuse or malicious behaviour.

To improve system performance and security, it is recommended to focus on reducing failure rates through better validation, investigate high-traffic periods, monitor suspicious IPs, and consider optimizing server performance to handle peak loads more effectively.