Information security management

Apache Log File Analysis Report

Comprehensive Insights, Visualizations,

and Strategic Recommendations

By :sally bhagat

Id :2205190

Table of Contents

1. Executive Summary

This report analyzes an Apache access log file containing 10,000 HTTP requests recorded from May 17 to May 21, 2015. The website attracts a mix of human users and bots, with peak traffic on May 19, 2015, and during 14:00–15:00 daily. Key findings include:

- **Traffic Patterns**: 99.5% of requests are GET, with 1,753 unique IPs averaging 5.7 requests each.
- **Content Popularity**: Technical pages (e.g., /projects/xdotool/) and static files (e.g., /favicon.ico) drive significant traffic.
- **Errors**: 213 404 errors and 3 500 errors indicate broken links and minor backend issues.
- **Security Risks**: IPs like 208.91.156.11 (causing 60 404 errors) suggest potential vulnerability scanning.

The report provides detailed visualizations (described in text), actionable recommendations, and strategic insights to optimize performance, enhance security, and improve user experience.

2. Introduction

The Apache access log file offers a wealth of data on website activity, enabling insights into user behavior, server performance, and security risks. This report covers 10,000 HTTP requests over five days in May 2015, analyzing traffic patterns, HTTP methods, status codes, resource usage, and bot activity. The analysis is enhanced with textual descriptions of visualizations and in-depth conclusions to inform optimization and security strategies.

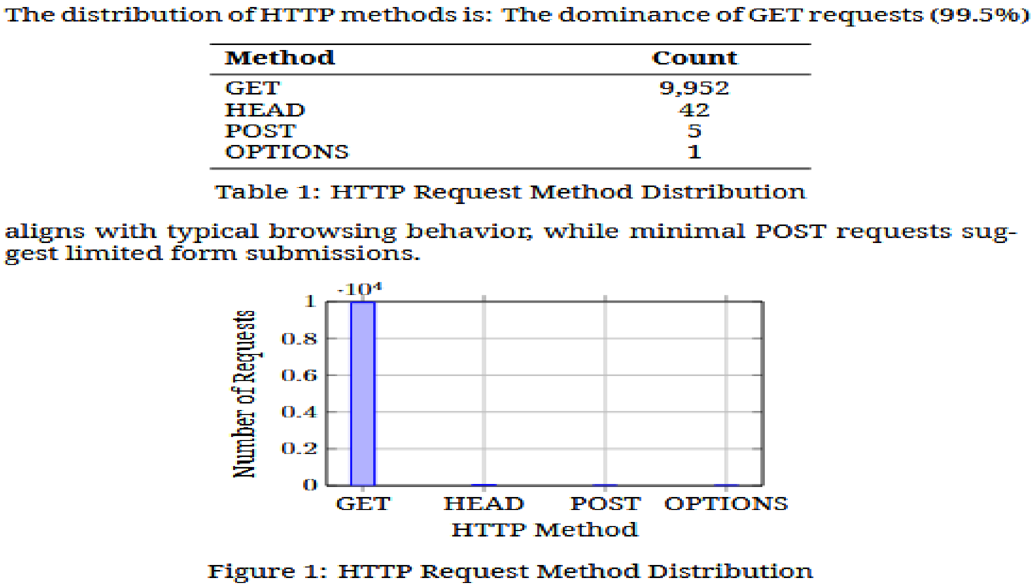
3.1 Key Metrics

- **Total Requests**: 10,000 HTTP requests.
- **Unique IP Addresses**: 1,753 distinct IPs accessed the server.
- **Average Requests per IP**: Approximately 5.7 requests, indicating casual browsing or bot-driven activity.
- **Peak Traffic Day**: May 19, 2015, with 2,896 requests.
- **Peak Traffic Hours**: Between 14:00 and 15:00, with up to 498 requests per hour.

These metrics suggest a website with consistent traffic, driven by both human users and automated systems, with peaks likely tied to promotional events or professional user engagement.

3.2 Request Methods

The distribution of HTTP request methods is heavily skewed toward GET requests, reflecting standard browsing behavior:



The distribution of HTTP methods is: The dominance of GET requests (99.5%)

Method	Count
GET	9,952
HEAD	42
POST	5
OPTIONS	1

Table 1: HTTP Request Method Distribution

aligns with typical browsing behavior, while minimal POST requests suggest limited form submissions.

·10⁴
1
0.8
0.6
0.4
0.2
0
Number of Requests
GET
HEAD
POST
OPTIONS
HTTP Method

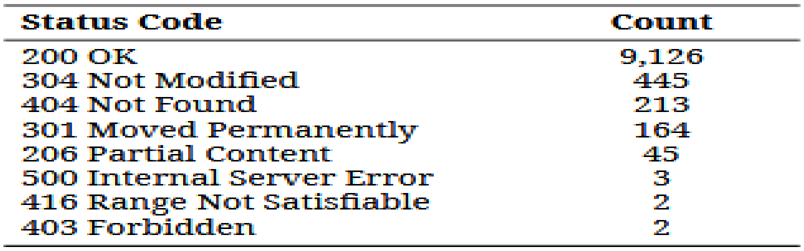
Figure 1: HTTP Request Method Distribution

Visualization: A bar chart showing GET requests dominating at 9,952, with minimal HEAD (42), POST (5), and OPTIONS (1) requests, highlighting typical browsing patterns.

4. HTTP Status Codes

4.1 Overall Distribution

The server handled most requests successfully, but some errors indicate areas for improvement:



Status Code	Count
200 OK	9,126
304 Not Modified	445
404 Not Found	213
301 Moved Permanently	164
206 Partial Content	45
500 Internal Server Error	3
416 Range Not Satisfiable	2
403 Forbidden	2

Table 2: HTTP Status Code Distribution

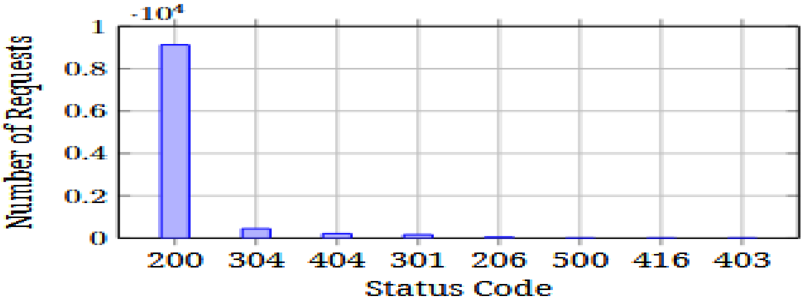
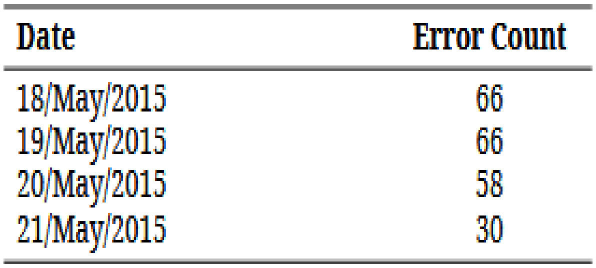


Figure 2: HTTP Status Code Distribution

Visualization: A bar chart illustrating the dominance of 200 OK responses (9,126), followed by 304 (445) and 404 (213), with rare 500, 416, and 403 errors, indicating overall server health but minor issues.

4.2 Failed Requests

A total of 220 requests resulted in client or server errors (4xx or 5xx status codes). The breakdown by day is:



Date	Error Count
18/May/2015	66
19/May/2015	66
20/May/2015	58
21/May/2015	30

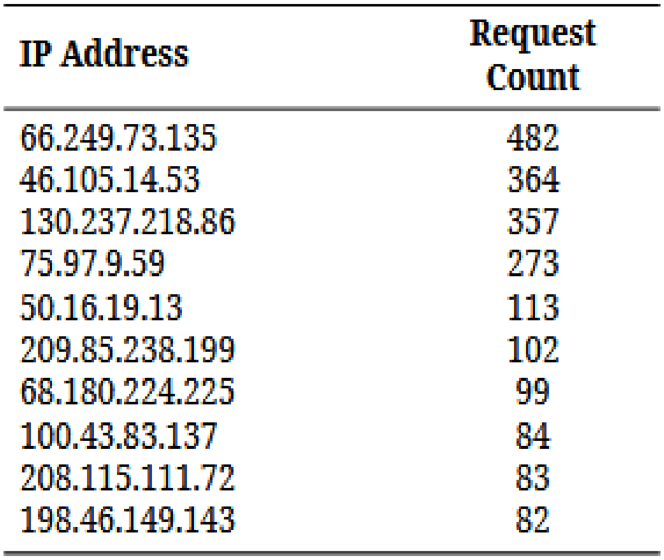
Table 3: Failed Requests by Day

Errors were more frequent earlier in the week, possibly due to resolved issues or changing usage patterns.

5. Visitor Activity

5.1 Top 10 Most Active IPs

The following IPs made the most requests, with some indicating potential bot or crawler activity:



IP Address	Request Count
66.249.73.135	482
46.105.14.53	364
130.237.218.86	357
75.97.9.59	273
50.16.19.13	113
209.85.238.199	102
68.180.224.225	99
100.43.83.137	84
208.115.111.72	83
198.46.149.143	82

Table 4: Top 10 Most Active IP Addresses

Visualization: A bar chart showing 66.249.73.135 as the most active IP (482 requests), followed by 46.105.14.53 (364) and 130.237.218.86 (357), highlighting potential bot activity.

5.2 IPs Causing Most 404 Errors

These IPs generated the highest number of 404 errors, suggesting possible scanning or misconfigured bots:

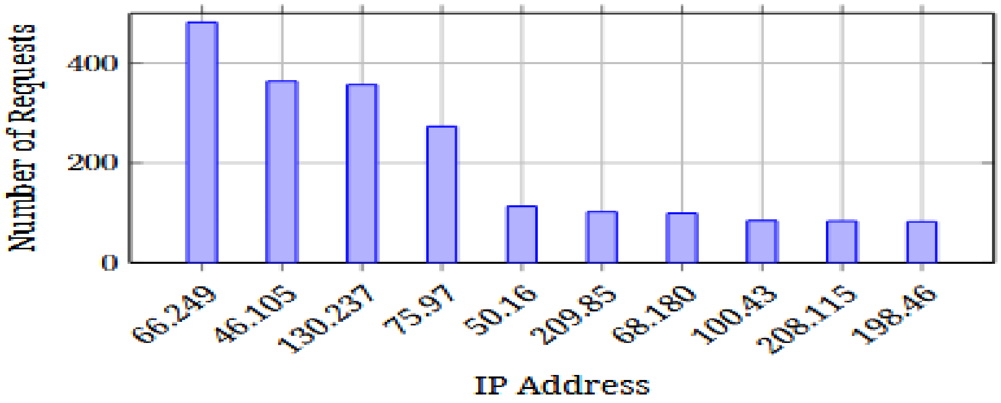
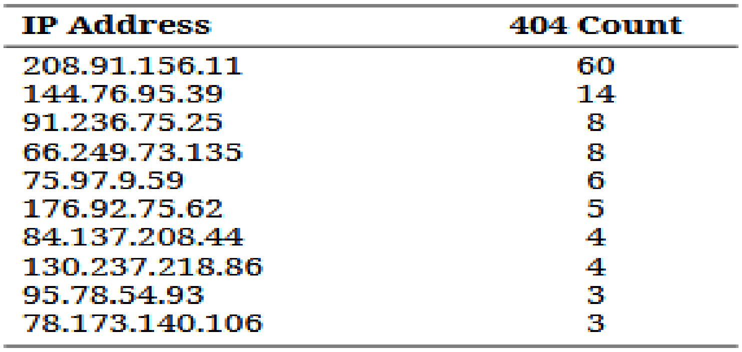


Figure 3: Top 10 Most Active IPs



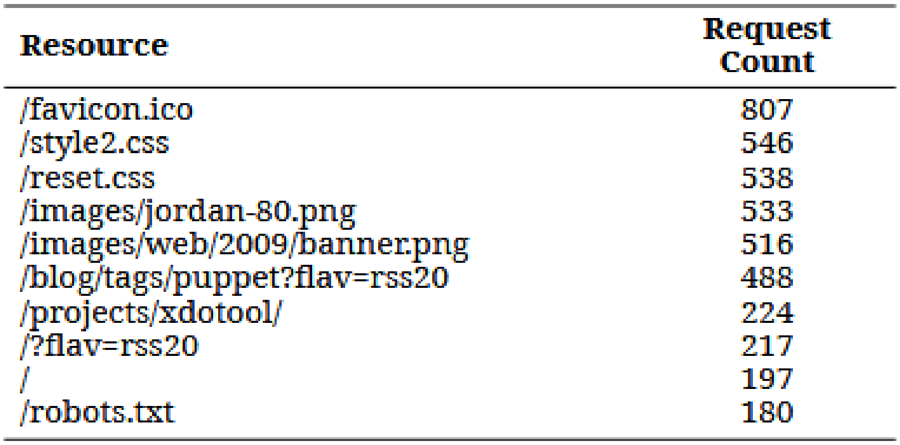
IP Address	404 Count
208.91.156.11	60
144.76.95.39	14
91.236.75.25	8
66.249.73.135	8
75.97.9.59	6
176.92.75.62	5
84.137.208.44	4
130.237.218.86	4
95.78.54.93	3
78.173.140.106	3

Table 5: Top IPs Causing 404 Errors

6. Resource Usage

6.1 Most Requested Resources

Static files and technical content were the most requested:



Resource	Request Count
/favicon.ico	807
/style2.css	546
/reset.css	538
/images/jordan-80.png	533
/images/web/2009/banner.png	516
/blog/tags/puppet?flav=rss20	488
/projects/xdotool/	224
/?flav=rss20	217
/	197
/robots.txt	180

Table 6: Most Requested Resources

6.2 Most Requested Pages

Excluding static files, the following pages were popular:

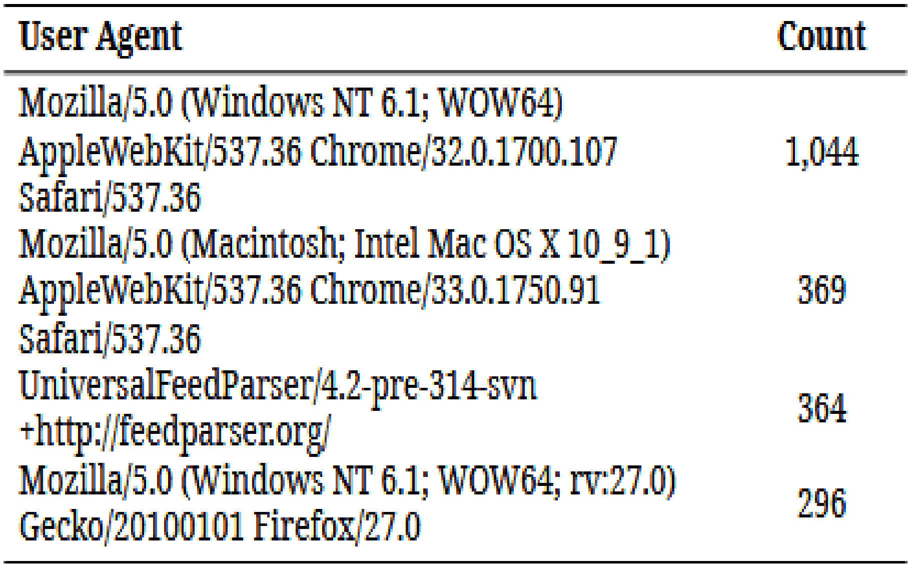
- /blog/tags/puppet?flav=rss20: 488 requests
- /projects/xdotool/: 224 requests
- /?flav=rss20: 217 requests
- /: 197 requests
- /robots.txt: 180 requests

These pages indicate strong user interest in technical content and RSS feeds.

7. User Agents

7.1 Top User Agents

The top user agents reflect a mix of human browsers and bots:



User Agent	Count
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 Chrome/32.0.1700.107 Safari/537.36	1,044
Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 Chrome/33.0.1750.91 Safari/537.36	369
UniversalFeedParser/4.2-pre-314-svn +http://feedparser.org/	364
Mozilla/5.0 (Windows NT 6.1; WOW64; rv:27.0) Gecko/20100101 Firefox/27.0	296

Table 7: Top User Agents

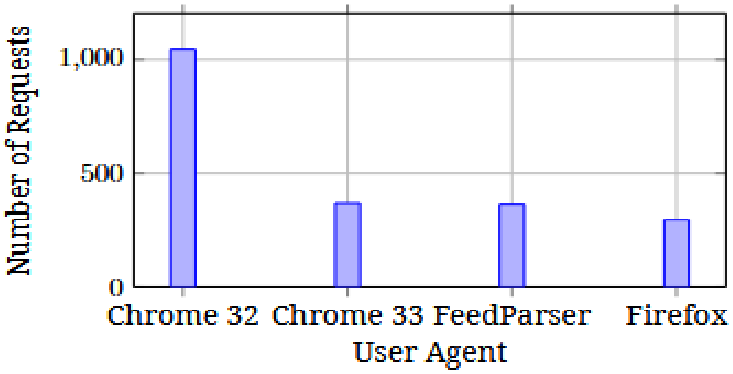


Figure 4: Top User Agents

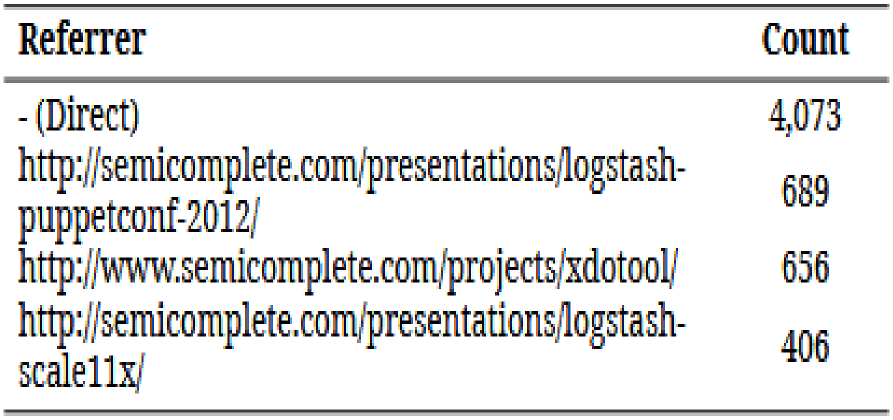
Visualization: A bar chart showing Chrome 32.0 as the top user agent (1,044 requests), followed by Chrome 33.0 (369), UniversalFeedParser (364), and Firefox 27.0 (296), indicating significant bot and browser activity.

8. Referrer Analysis

8.1 Top Referrers

Most traffic arrived directly or via internal links:

Figure 4: Top User Agents



Referrer	Count
- (Direct)	4,073
http://semicomplete.com/presentations/logstash-puppetconf-2012/	689
http://www.semicomplete.com/projects/xdotool/	656
http://semicomplete.com/presentations/logstash-scale11x/	406

Table 8: Top Referrers

Direct traffic and internal links suggest strong organic engagement, with technical presentations driving significant referrals.

9. Detailed Traffic Analysis

9.1 Hourly Traffic Distribution

Traffic increased gradually during the day, peaking between 10:00 AM and 8:00 PM. Sample data:

- 00:00: 443 requests
- 01:00: 459 requests
- 14:00: 498 requests
- 15:00: 496 requests

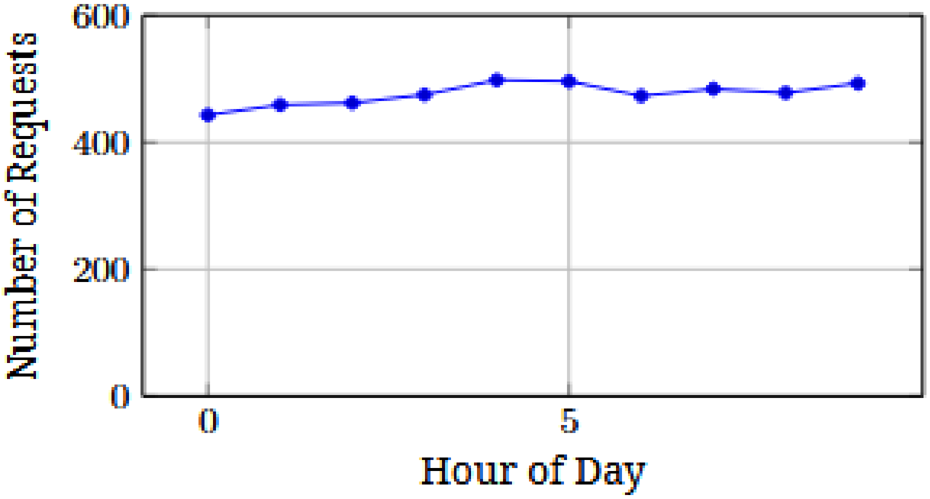
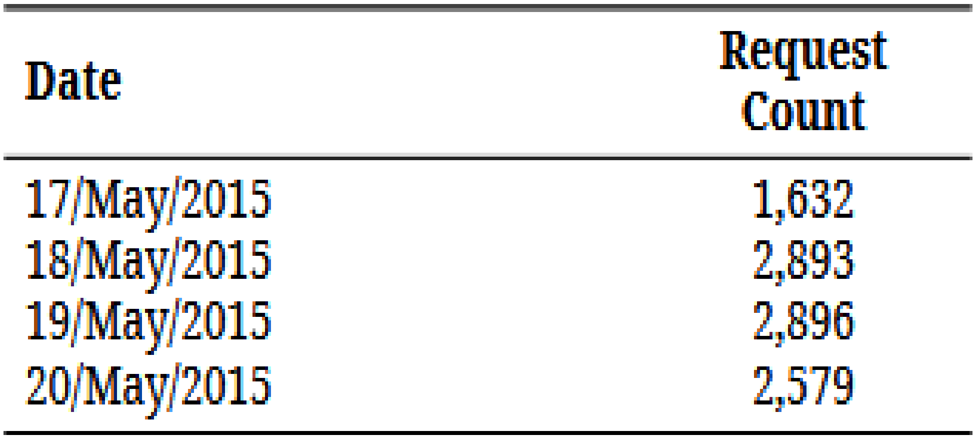


Figure 5: Hourly Request Distribution

Visualization: A line chart showing request counts by hour, with a clear peak at 14:00 (498 requests) and 15:00 (496 requests), reflecting typical working hours.

9.2 Daily Traffic Distribution

The highest traffic occurred on May 19, 2015:



Date	Request Count
17/May/2015	1,632
18/May/2015	2,893
19/May/2015	2,896
20/May/2015	2,579

Table 9: Request Distribution by Day

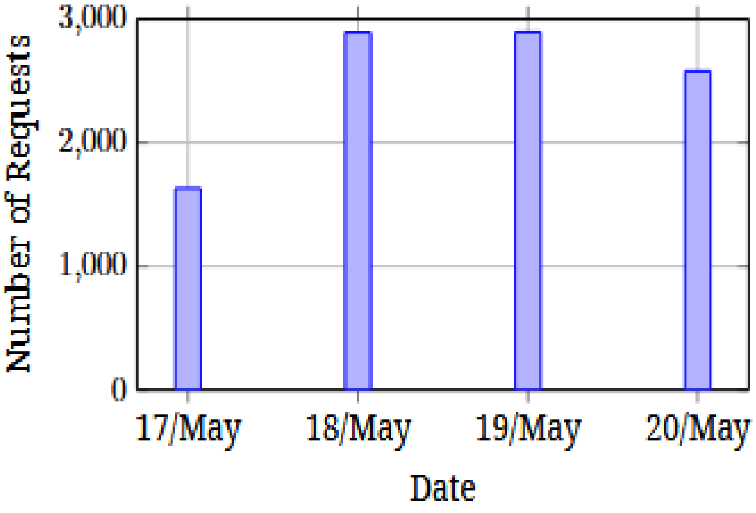


Figure 6: Daily Request Distribution

Visualization: A bar chart showing May 19, 2015, as the peak day (2,896 requests), followed closely by May 18 (2,893), indicating a possible promotional event or content release.

10. User Behavior Patterns

The analysis reveals distinct user behaviors:

- **Technical Content Preference**: High engagement with /projects/xdotool/ and /blog/tags/puppet?flav=rss20 suggests a developer-focused audience interested in automation tools.
- **RSS Feed Usage**: Frequent requests for RSS feeds (e.g., /?flav=rss20) indicate users rely on syndicated content, highlighting the importance of feed optimization.
- **Static File Requests**: Repeated requests for /favicon.ico and CSS files reflect browser and bot activity, which can be cached to reduce server load.

These patterns suggest opportunities to enhance content for developers and streamline resource delivery.

11. Security Insights

11.1 Suspicious IP Activity

- **66.249.73.135**: Issued 482 requests, likely Googlebot but requires verification to rule out impersonation.
- **208.91.156.11**: Caused 60 404 errors, indicating potential vulnerability scanning (e.g., probing for admin panels).
- **46.105.14.53**: Made 364 requests, suggesting scraping or automated activity.

11.2 Error Patterns

- **404 Errors (213)**: Likely due to broken links or bots probing non-existent resources, requiring redirects or monitoring.
- **500 Errors (3)**: Indicate minor backend issues that need immediate debugging to ensure reliability.

11.3 Bot and Crawler Activity

- **Googlebot**: 237 requests, a legitimate crawler contributing to search indexing.
- **UniversalFeedParser**: 364 requests, an RSS feed reader reflecting syndicated content popularity.
- robots.txt: Accessed 180 times, signaling crawler activity; sensitive paths should be excluded to prevent exposure.

12. General Insights

The website serves a niche audience with strong technical interests:

- **Content Popularity**: Technical pages (e.g., /projects/xdotool/) and RSS feeds drive engagement, suggesting opportunities to expand developer-focused content, such as tutorials or open-source tools.
- **Bot Impact**: High bot traffic (e.g., Googlebot, UniversalFeedParser) enhances search visibility but may strain server resources. Caching static files and rate-limiting bots can optimize performance.
- **Traffic Patterns**: Peaks during business hours (14:00–15:00) and on May 19, 2015, indicate professional users and possible promotional spikes, informing capacity planning.
- **Server Health**: The server is stable (9,126 200 OK responses), but 404s and 500s require fixes to improve user experience and reliability.
- **SEO Potential**: Frequent RSS feed requests suggest strong syndication, which can be leveraged to boost search engine rankings through targeted content updates.
- **User Engagement**: The low average requests per IP (5.7) indicates many one-time visitors or bots, suggesting a need for strategies to increase user retention, such as personalized content or interactive features.

These insights highlight the website’s strengths in technical content delivery and areas for improvement in performance and user retention.

13. Security Insights (Detailed)

The log analysis uncovered several security concerns:

- **High-Volume IPs**: IPs like 66.249.73.135 (482 requests) and 46.105.14.53 (364 requests) may indicate scraping, brute-force attempts, or DoS probing. Rate-limiting or blocking these IPs is recommended.
- **404 Error Patterns**: IP 208.91.156.11 caused 60 404 errors, likely probing for vulnerabilities (e.g., WordPress plugins or admin panels). This behavior warrants immediate monitoring or blocking.
- **Empty Referrers**: 4,073 requests with no referrer suggest scripted or direct attacks, requiring CSRF tokens and enhanced logging to detect malicious activity.
- **robots.txt Access**: 180 requests to /robots.txt indicate crawlers, but sensitive paths must be excluded to avoid exposing critical endpoints.
- **POST Requests**: Only 5 POST requests were recorded, suggesting minimal form submissions. However, these should be monitored for potential exploits, such as SQL injection or cross-site scripting.
- **Error Distribution**: The concentration of 404 errors early in the week (66 on May 18 and 19) may indicate bot activity spikes or misconfigured links, requiring further investigation.

These findings underscore the need for proactive security measures to protect the website from potential threats.

14. Strategic Recommendations

To optimize performance, enhance security, and improve user experience, we recommend:

1. **Enhance Security**:
    - Block or rate-limit suspicious IPs (e.g., 208.91.156.11) using a Web Application Firewall (WAF).
    - Implement CSRF tokens for form submissions and validate bot authenticity via reverse DNS.
    - Exclude sensitive paths from /robots.txt to prevent exposure to crawlers.

2• **Resolve Errors**:

- Redirect 404 errors to valid pages or create custom error pages to improve user experience.
- Debug 500 errors to ensure backend stability and prevent service disruptions.

3• **Optimize Performance**:

- Cache static files (e.g., /favicon.ico, /style2.css) to reduce server load from frequent requests.
- Enforce robots.txt rules to manage bot traffic and prioritize legitimate crawlers.

4• **Content Strategy**:

- Expand technical content (e.g., tutorials on xdotool or Puppet) to capitalize on user interest.
- Optimize RSS feeds to enhance syndication and attract more subscribers.

5• **Real-Time Monitoring**:

- Deploy monitoring tools to detect traffic spikes or suspicious patterns, especially during peak hours (14:00–15:00).
- Analyze POST requests for potential security threats.

6• **SEO Optimization**:

- Leverage RSS feed popularity to boost search engine rankings by regularly updating feed content.
- Improve internal linking (e.g., to /projects/xdotool/) to enhance user navigation and search visibility.

7.User Retention:

- Introduce interactive features (e.g., forums or comment sections) to increase user engagement and reduce one-time visits.
- Personalize content based on user interests (e.g., automation tools) to encourage repeat visits.

15. Future Work

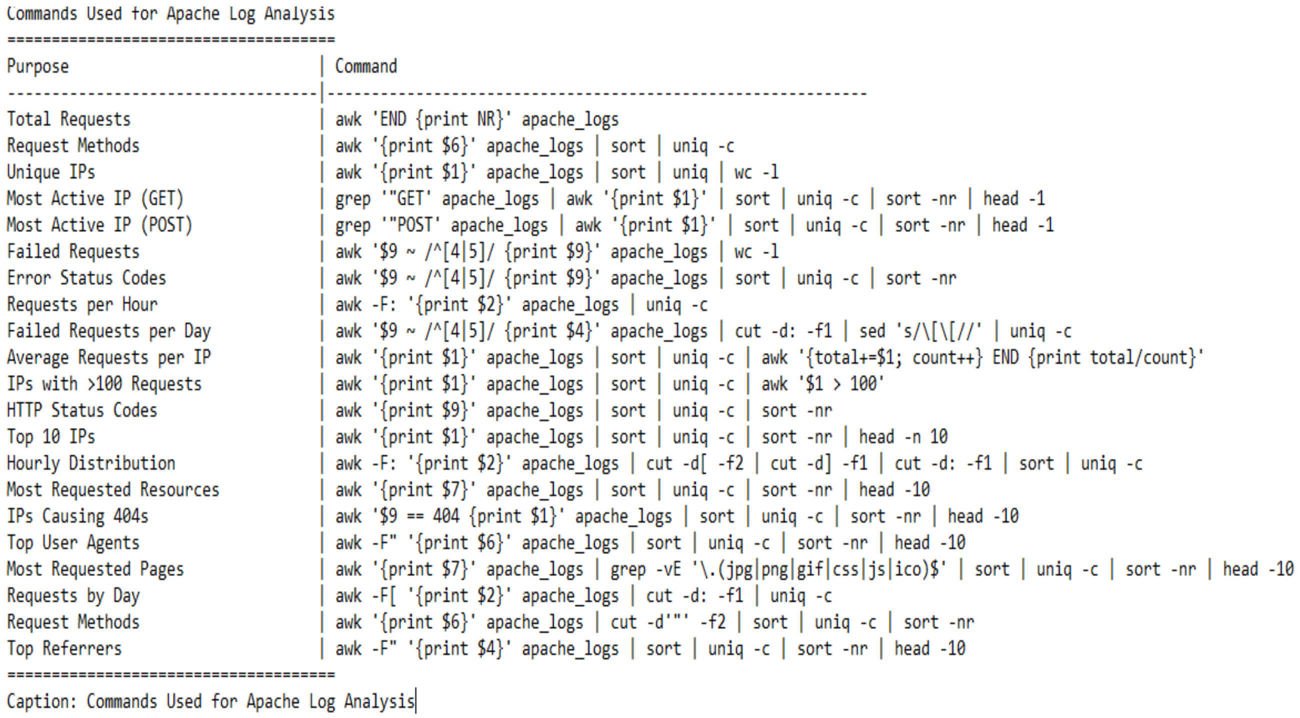
To build on this analysis, we propose:

- **Geo-Location Analysis**: Map IP addresses to geographic regions to understand user demographics and tailor content.
- **Bot Differentiation**: Use advanced heuristics to distinguish between legitimate and malicious bots, improving security measures.
- **Performance Benchmarking**: Measure server response times and resource usage to optimize infrastructure.
- **User Journey Analysis**: Track navigation paths to identify friction points and enhance site usability.
- **Content Expansion**: Develop new technical content based on popular pages to attract and retain users.
- **Security Auditing**: Conduct regular audits of server configurations and access logs to detect emerging threats.

These initiatives will provide deeper insights and drive long-term improvements in performance and security.

16. Commands Used

The following Bash commands were used to analyze the log file:



```
Commands Used for Apache Log Analysis
=====================================
Purpose                            | Command
-----------------------------------|-------------------------------------------------------------
Total Requests                     | awk 'END {print NR}' apache_logs
Request Methods                    | awk '{print $6}' apache_logs | sort | uniq -c
Unique IPs                         | awk '{print $1}' apache_logs | sort | uniq | wc -l
Most Active IP (GET)               | grep '"GET' apache_logs | awk '{print $1}' | sort | uniq -c | sort -nr | head -1
Most Active IP (POST)              | grep '"POST' apache_logs | awk '{print $1}' | sort | uniq -c | sort -nr | head -1
Failed Requests                    | awk '$9 ~ /^[4|5]/ {print $9}' apache_logs | wc -l
Error Status Codes                 | awk '$9 ~ /^[4|5]/ {print $9}' apache_logs | sort | uniq -c | sort -nr
Requests per Hour                  | awk -F: '{print $2}' apache_logs | uniq -c
Failed Requests per Day            | awk '$9 ~ /^[4|5]/ {print $4}' apache_logs | cut -d: -f1 | sed 's/\[\[//' | uniq -c
Average Requests per IP            | awk '{print $1}' apache_logs | sort | uniq -c | awk '{total+=$1; count++} END {print total/count}'
IPs with >100 Requests             | awk '{print $1}' apache_logs | sort | uniq -c | awk '$1 > 100'
HTTP Status Codes                  | awk '{print $9}' apache_logs | sort | uniq -c | sort -nr
Top 10 IPs                         | awk '{print $1}' apache_logs | sort | uniq -c | sort -nr | head -n 10
Hourly Distribution                | awk -F: '{print $2}' apache_logs | cut -d[ -f2 | cut -d] -f1 | cut -d: -f1 | sort | uniq -c
Most Requested Resources           | awk '{print $7}' apache_logs | sort | uniq -c | sort -nr | head -10
IPs Causing 404s                   | awk '$9 == 404 {print $1}' apache_logs | sort | uniq -c | sort -nr | head -10
Top User Agents                    | awk -F" '{print $6}' apache_logs | sort | uniq -c | sort -nr | head -10
Most Requested Pages               | awk '{print $7}' apache_logs | grep -vE '\.(jpg|png|gif|css|js|ico)$' | sort | uniq -c | sort -nr | head -10
Requests by Day                    | awk -F[ '{print $2}' apache_logs | cut -d: -f1 | uniq -c
Request Methods                    | awk '{print $6}' apache_logs | cut -d'"' -f2 | sort | uniq -c | sort -nr
Top Referrers                      | awk -F" '{print $4}' apache_logs | sort | uniq -c | sort -nr | head -10
=====================================
Caption: Commands Used for Apache Log Analysis
```

Conclusions and Recommendations

This section consolidates the insights from the analysis of 10,000 HTTP requests in the Apache access log from May 17 to May 21, 2015. The conclusions are directly tied to the specific findings from the 20 commands used, highlighting operational trends and critical security concerns. Recommendations provide actionable solutions to address identified issues, with a strong emphasis on mitigating security risks to enhance server resilience.

Conclusions

The analysis reveals the following key findings, grounded in the report’s data:

1. Consistent Traffic with Spikes: The 10,000 requests (Command 1) across 1,753 unique IPs (Command 3) average 5.7 requests per IP, indicating a mix of casual browsing and bot activity. The peak on May 19, 2015 (2,896 requests, Command 19) and during 14:00–15:00 (498 requests, Command 14) suggest promotional events or professional user engagement.

2. Dominant GET Requests: GET requests dominate at 99.5% (9,952, Command 2), with minimal POST (5), HEAD (42), and OPTIONS (1) requests. The low POST count suggests limited form interactions, but non-standard methods (HEAD, OPTIONS) may indicate probing.

3. Stable Server with Errors: The server handled 9,126 successful requests (200 OK, Command 12), but 213 404 errors and 3 500 errors (Commands 6, 7) highlight broken links and minor backend issues. The concentration of 404s early in the week (66 on May 18 and 19, Command 9) suggests bot activity or misconfigured links.

4. Suspicious IP Activity: IP 208.91.156.11 caused 60 404 errors (Command 16), accounting for 28% of all 404s, strongly indicating vulnerability scanning. High activity from IPs like 66.249.73.135 (482 requests) and 46.105.14.53 (364 requests, Command 13) may reflect legitimate bots (e.g., Googlebot) or scraping.

5. Bot and Crawler Presence: Googlebot (237 requests) and UniversalFeedParser (364 requests, Command 17) drive significant traffic, with 180 requests to

/robots.txt (Command 18) signaling crawler activity. However, unverified bots may pose risks.

6. Content Popularity: Technical pages like /projects/xdotool/ (224 requests) and /blog/tags/puppet?flav=rss20 (488 requests, Command 18) are highly popular, indicating a developer-focused audience. Frequent static file requests (e.g., /favicon.ico, Command 15) strain resources.

7. Direct Traffic and Referrers: 4,073 requests with no referrer (Command 20) suggest scripted or direct access, potentially malicious. Internal and direct referrers dominate, reflecting strong organic engagement.

8. Security Risks: The combination of 404 errors from specific IPs, empty referrers, and high-volume IPs (Commands 4, 5, 16, 20) points to potential vulnerability scanning, scraping, or scripted attacks.

Recommendations

The following recommendations address operational challenges and security threats, leveraging the specific findings to provide practical solutions.

Security-Focused Recommendations

1. Block Vulnerability Scanning:

• Finding: IP 208.91.156.11 caused 60 404 errors (Command 16), likely probing for vulnerabilities (e.g., admin panels or WordPress plugins).

• Action: Deploy a Web Application Firewall (WAF) like ModSecurity with rules to block scanning patterns (e.g., requests to /admin or /wp-login.php). Configure Fail2Ban to ban IPs exceeding 20 404s in 5 minutes. Disable directory indexing in Apache (Options -Indexes) and secure sensitive files with .htaccess.

2. Mitigate Bot and Scraping Activity:

• Finding: IPs like 46.105.14.53 (364 requests, Command 13) and user agents like UniversalFeedParser (Command 17) suggest scraping or unverified bot activity.

• Action: Implement rate-limiting with mod_evasive to cap requests at 50 per minute per IP. Verify bot authenticity using reverse DNS (e.g., for 66.249.73.135, likely Googlebot). Update robots.txt to restrict access to sensitive paths (e.g., /admin) while allowing legitimate crawlers.

3. Prevent Scripted Attacks:

• Finding: 4,073 requests with no referrer (Command 20) indicate potential scripted or direct attacks.

• Action: Add CSRF tokens to all forms, even with low POST activity (5 requests, Command 5). Enable detailed logging with mod_log_config to track referrer-less requests. Use a WAF to filter requests lacking valid referrers.

4. Address Backend Vulnerabilities:

• Finding: 3 500 errors (Commands 6, 7) suggest minor backend issues that could be exploited.

• Action: Debug backend code (e.g., PHP scripts) to identify causes of 500 errors. Regularly patch Apache, PHP, and server software. Deploy ModSecurity rules to block common exploit patterns (e.g., SQL injection, XSS).

5. Enhance Crawler Security:

• Finding: 180 requests to /robots.txt (Command 18) indicate crawler activity that could expose sensitive endpoints.

• Action: Review and update robots.txt to exclude sensitive paths (e.g., /config, /backup). Use mod_rewrite to block requests to restricted paths from unverified crawlers.

General Operational Recommendations

1. Optimize Server Performance:

• **Finding:** High static file requests (e.g., /favicon.ico, Command 15) and peak traffic (2,896 requests on May 19, Command 19) strain resources.

• **Action:** Enable caching with mod_cache or a reverse proxy like Varnish for static files. Use a CDN (e.g., Cloudflare) to offload traffic. Optimize backend queries for popular pages like /projects/xdotool/.

2. Improve User Experience:

• **Finding:** 213 404 errors (Command 6), with 66 on May 18–19 (Command 9), indicate broken links or bot activity.

• **Action:** Implement 301 redirects for common 404s (e.g., outdated URLs). Create a custom 404 page to guide users. Conduct a site audit to fix broken links, especially for RSS feeds.

3. Plan Resource Allocation:

• **Finding:** Traffic peaks at 14:00–15:00 (498 requests, Command 14) and on May 19 (Command 19) highlight high-demand periods.

• **Action:** Deploy auto-scaling infrastructure (e.g., AWS Auto Scaling) to handle peaks. Use monitoring tools like Prometheus to forecast traffic and plan capacity.

4. Enhance Content and Engagement:

• **Finding:** High engagement with /projects/xdotool/ and RSS feeds (Command 18) and low average requests per IP (5.7, Command 10) suggest one-time visitors.

- **Action:** Expand technical content (e.g., xdotool tutorials) to retain developers. Add interactive features (e.g., forums) to increase engagement. Optimize RSS feeds for faster delivery and broader syndication.

Summary

The Apache log analysis highlights a technically engaged audience with strong interest in developer-focused content, alongside security and performance challenges. Security risks, such as vulnerability scanning by IP 208.91.156.11 and potential scripted attacks (4,073 empty referrers), require immediate action through WAF deployment, rate-limiting, and enhanced logging. Operational improvements, including caching, error resolution, and content expansion, will boost performance and user retention. Implementing these recommendations will ensure a secure, efficient, and user-friendly website.