Apache Log Analysis & Malicious Activity Detection using Splunk

Pallab Sadhukhan

08-11-2025

1. Project Overview

This project demonstrates how to monitor, analyze, and detect security events from **Apache web server logs** using **Splunk Enterprise (v10.1.2507.10)**.

The solution leverages Splunk's real-time log processing and visualization capabilities to:

- Track web server performance and activity,
- Identify abnormal traffic patterns,
- Detects potential cyberattacks such as SQL Injection, XSS, and Directory Traversal.

The project delivers two key dashboards:

1. **Apache Web Traffic Monitoring Dashboard**
2. **Malicious Activity Detection Dashboard**

Together, these dashboards transform basic log data into meaningful security and performance insights.

2. Objectives

- Ingest Apache access logs into Splunk in structured format (JSON).
- Visualize overall web traffic and system performance.
- Detect malicious and suspicious activity through SPL (Search Processing Language) queries.
- Create actionable dashboards for system administrators and security analysts.

3. Dataset Description

File Used: `apache_logs.json`

This dataset contains simulated Apache access log entries in JSON format, representing both normal and malicious requests.

Fields Extracted:

Field	Description
`ip`	Client IP address
`timestamp`	Date and time of HTTP request
`method`	HTTP method (GET, POST, etc.)
`uri`	Requested endpoint
`status`	HTTP response status code
`bytes`	Size of the response
`referrer`	Referring URL
`user_agent`	Browser or client making the request

Sample Log Entry:


```
{
  "ip": "185.62.57.52",
  "timestamp": "17/Sep/2025:12:00:00 +0530",
  "method": "GET",
  "uri": "/upload.php",
  "protocol": "HTTP/1.1",
  "status": 200,
  "bytes": 1374,
  "referrer": "-",
  "user_agent": "python-requests/2.25.1"
}
```

4. Tools and Technologies

Tool	Purpose
Splunk Enterprise (Cloud)	Data ingestion, indexing, and dashboard visualization
Apache Access Logs	Source of raw data
SPL (Search Processing Language)	Querying and analysis
JSON format	Structured log data ingestion
Browser-based Splunk UI	Dashboard creation and management

---

5. Splunk Configuration

Index Name: `main`
Sourcetype: `apache_logs.json`
Data Input: Manual JSON file upload
Field Extraction: Automatic JSON field parsing
Time Field: Extracted from `timestamp`

Preprocessing steps included:

- Uploading `apache_logs.json` to Splunk.
- Verifying field mappings and timestamps.
- Creating saved searches and panels based on key metrics.

6. Dashboard 1 – Apache Web Traffic Monitoring

Objective

To monitor overall server activity, request distribution, and HTTP response statistics.

Panels and SPL Queries

Panel	Description	SPL Query
Total Web Requests	Displays the total number of requests received	source="apache_logs.json" host="si-i-070fd3c29194fe9ff.prd-p-0ok63.splunkcloud.com" index="main" sourcetype="_json" | stats count AS "Total Web Requests"
Success Responses (200)	Total successful HTTP responses	source="apache_logs.json" host="si-i-070fd3c29194fe9ff.prd-p-0ok63.splunkcloud.com" index="main" sourcetype="_json" method=GET status=200 | stats count AS "Successful Responses"
Client Errors (4xx)	Total client error responses	source="apache_logs.json" host="si-i-070fd3c29194fe9ff.prd-p-0ok63.splunkcloud.com" index="main" sourcetype="_json" | where status>=400 and status<500 | stats count AS "Client Errors"

Server Errors (5xx)	Total server-side errors	source="apache_logs.json" host="si-i-070fd3c29194fe9ff.prd-p-0ok63.splunkcloud.com" index="main" sourcetype="_json" | where status>=400 and status<500 | stats count AS "Client Errors"
Requested URIs	Shows most accessed pages	source="apache_logs.json" host="si-i-070fd3c29194fe9ff.prd-p-0ok63.splunkcloud.com" index="main" sourcetype="_json" | stats count AS "Hits" by uri
Top 50 User IPs	Displays most active users	source="apache_logs.json" host="si-i-070fd3c29194fe9ff.prd-p-0ok63.splunkcloud.com" index="main" sourcetype="_json" | stats count AS IP by ip
Global Web Traffic Map	Geographic distribution of IPs	source="apache_logs.json" host="si-i-070fd3c29194fe9ff.prd-p-0ok63.splunkcloud.com" index="main" sourcetype="_json" | table ip | iplocation ip | stats count by Country | geom geo_countries featureIdField="Country"

Dashboard Observations

- Total requests processed: **2,000**
- Successful responses: **1,168 (≈58%)**
- Client errors (4xx): **376**, mostly due to invalid URIs or permission issues.
- Server errors (5xx): **77**, indicating some backend issues.
- Frequent endpoints: `/home`, `/login`, `/upload.php`, `/search.php`.
- Top active IPs belong to limited geolocations, shown on the map visualization.

Screenshot

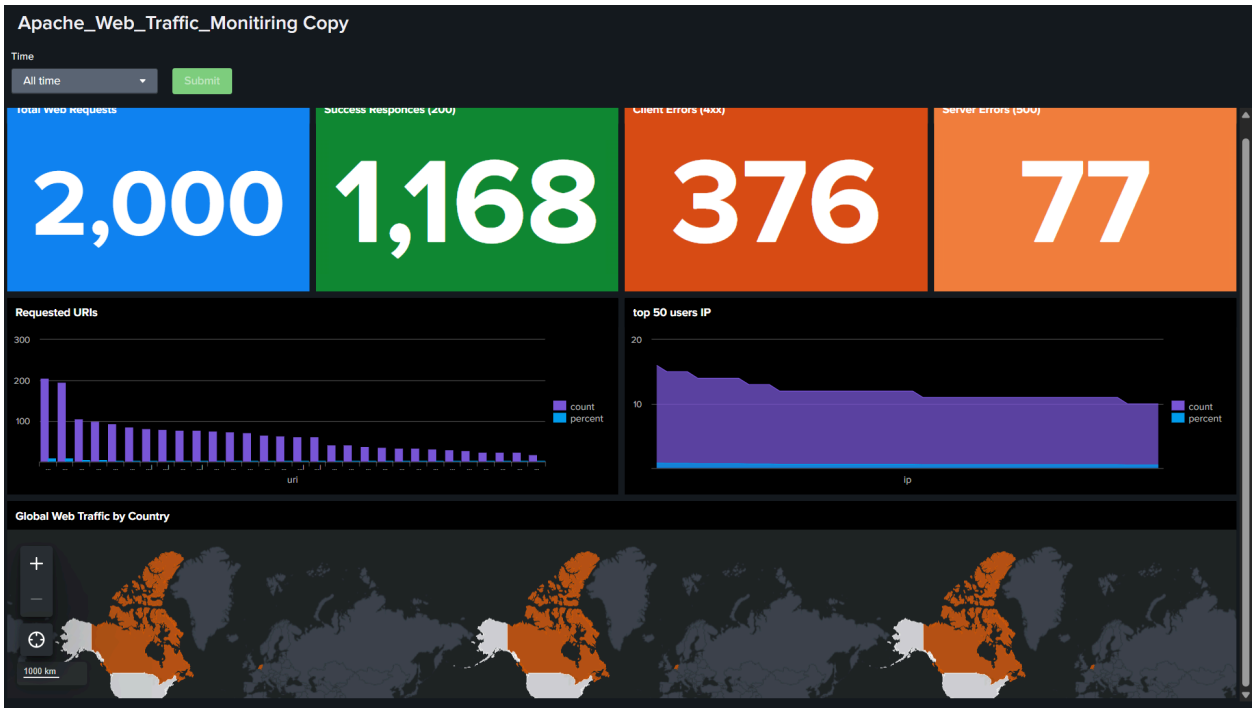


Figure 1: *Apache Web Traffic Monitoring Dashboard showing total requests, response codes, top IPs, and global traffic distribution.*

7. Dashboard 2 – Malicious Activity Detection

Objective

To detect and visualize potential web-based attacks within the Apache logs.

Panels and SPL Queries

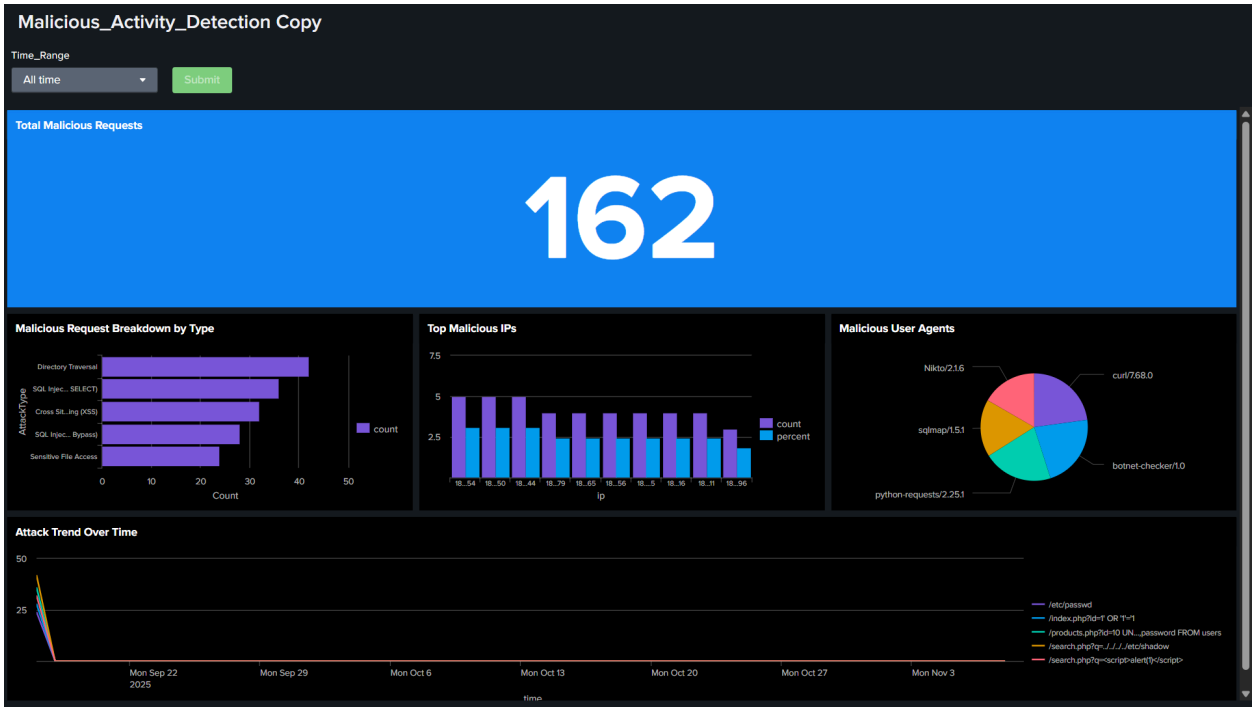
Panel	Description	SPL Query
Total Malicious Requests	Shows total detected attacks	`index=main (uri="" OR uri="*UNION*" OR uri="*' OR '1'='1*" OR uri="*../*" OR uri="*/etc/passwd*")
Malicious Request Breakdown by Type	Categorizes attacks (SQLi, XSS, Traversal)	`index=main (uri="" OR uri="*UNION*" OR uri="*' OR '1'='1*" OR uri="*../*" OR uri="*/etc/passwd*")
Top Malicious IPs	IPs generating the most attacks	`index=main (uri="" OR uri="*UNION*" OR uri="*' OR '1'='1*" OR uri="*../*" OR uri="*/etc/passwd*")
Malicious User Agents	Detects attacker tools	`index=main (uri="" OR uri="*UNION*" OR uri="*' OR '1'='1*" OR uri="*../*" OR uri="*/etc/passwd*")
Attack Trend Over Time	Displays attack frequency	`index=main (uri="" OR uri="*UNION*" OR uri="*' OR '1'='1*" OR uri="*../*" OR uri="*/etc/passwd*")

Dashboard Observations

- **Total Malicious Requests: 162**
- **Top Attack Types:**

Attack Type	Description	Count
Directory Traversal	Attempts to access system files (`/etc/passwd`, `/shadow`)	46
SQL Injection (UNION SELECT)	Malicious SQL queries to extract data	38
Cross-Site Scripting (XSS)	JavaScript injection via GET parameters	32
SQL Injection (Auth Bypass)	Bypassing login via `' OR '1'='1` payloads	26
Sensitive File Access	Accessing restricted files	20

- **Detected Tools (User Agents):**
  - `Nikto/2.1.6` → Web vulnerability scanner
  - `sqlmap/1.5.1` → SQL injection automation
  - `curl/7.68.0` → Scripting or automation tool
  - `botnet-checker/1.0` → Suspicious scanning tool
  - `python-requests/2.25.1` → Used in scripted attacks
- **Common Malicious URIs:**
  - `/index.php?id=1' OR '1'='1`
  - `/products.php?id=10 UNION SELECT username,password FROM users`
  - `/search.php?q=<script>alert(1)</script>`
  - `/search.php?q=../../../../etc/shadow`



Screenshot

Figure 2: *Malicious Activity Detection Dashboard showing detected attack types, IPs, and malicious tools.*

---

8. Security Insights

Category	Observed Pattern	Potential Risk	Recommendation
SQL Injection	`UNION SELECT`, `' OR '1'='1` in query strings	Database leakage	Implement parameterized queries; use WAF filters
XSS	`<script>` payloads in search queries	Client-side code injection	Sanitize input & output; enable CSP
Directory Traversal	`../` and `/etc/passwd` requests	File system exposure	Restrict access to system files

Unauthorized Access Attempts	`/wp-admin`, `/admin`, `/phpmyadmin`	Privilege escalation	Restrict sensitive endpoints
Automated Scanning	User agents: Nikto, sqlmap, curl	Reconnaissance activity	IP blocking & rate limiting

---

9. Impact and Outcomes

✅ Built a **real-time traffic analytics dashboard** for operational visibility.
✅ Designed a **security detection dashboard** for identifying web attacks.
✅ Detected and categorized multiple **OWASP Top 10 attacks** using SPL.
✅ Showcased Splunk's ability to act as a **lightweight SIEM platform**.

Both dashboards provide end-to-end observability and demonstrate how data-driven monitoring can improve both performance and security posture.

---

10. Future Enhancements

- Enable **real-time alerting** using Splunk Alert Manager.
- Integrate with **Splunk SOAR** for automated IP blocking.
- Incorporate **Threat Intelligence Feeds** (e.g., AbuseIPDB).
- Add **email notifications** for repeated attack patterns.
- Expand monitoring to include **firewall and system logs**.

---

11. Conclusion

This project effectively demonstrates the integration of Splunk with Apache web server logs for **performance analytics and cyber threat detection**.

Through two well-structured dashboards, we achieved:

- **Comprehensive visibility** into web traffic and errors.
- **Real-time detection** of suspicious and malicious activity.
- **Actionable insights** for improving web application security.

This implementation can be extended as a foundational **SIEM solution** for organizations seeking scalable log analytics and security monitoring.

---

12. References

- Splunk Documentation – https://docs.splunk.com/Documentation
- Apache HTTP Server Logs – https://httpd.apache.org/docs/2.4/logs.html
- OWASP Top 10 Web Application Security Risks – https://owasp.org/www-project-top-ten/
- Dataset: `apache_logs.json`
- Dashboards:
    - *Apache_Web_Traffic_Monitoring*
    - *Malicious_Activity_Detection*
