

Alexander Ticket

Note: This report has been sanitized for public sharing.

All internal IPs, hostnames, and Splunk URLs have been redacted or replaced with simulated values.

Report was originally prepared for Jira; internal console links are not publicly accessible. Query references shown for context

QRadar ID: **54708**

Description

Multiple Threat Vectors Directed at a Single Destination IP, Including Time-Based SQL Injection Attempts

Victim:

[internal web server] - redacted-domain.local

1 Encoded log:

"/admin/update-issue-bookdeails.php?rid=%28SELECT%20%28CASE%20WHEN%20%286210%3D6210%29%20THEN%207%20ELSE%20%28SELECT%208233%20UNION%20SELECT%204070%29%20END%29%29"

1 Decoded log:

"/admin/update-issue-bookdeails.php?rid=(SELECT (CASE WHEN (6210=6210) THEN 7 ELSE (SELECT 8233 UNION SELECT 4070) END))"

2 Encoded log:

"/admin/update-issue-bookdeails.php?rid=7%29%20AND%20%28SELECT%205449%20FROM%20%28SELECT%28SLEEP%285%29%29%29xhtk%29%20AND%20%283162%3D3162"

2 Decoded log:

"/admin/update-issue-bookdeails.php?rid=7) AND (SELECT 5449 FROM (SELECT(SLEEP(5)))xhtk) AND (3162=3162"

ATTACKER INFO:

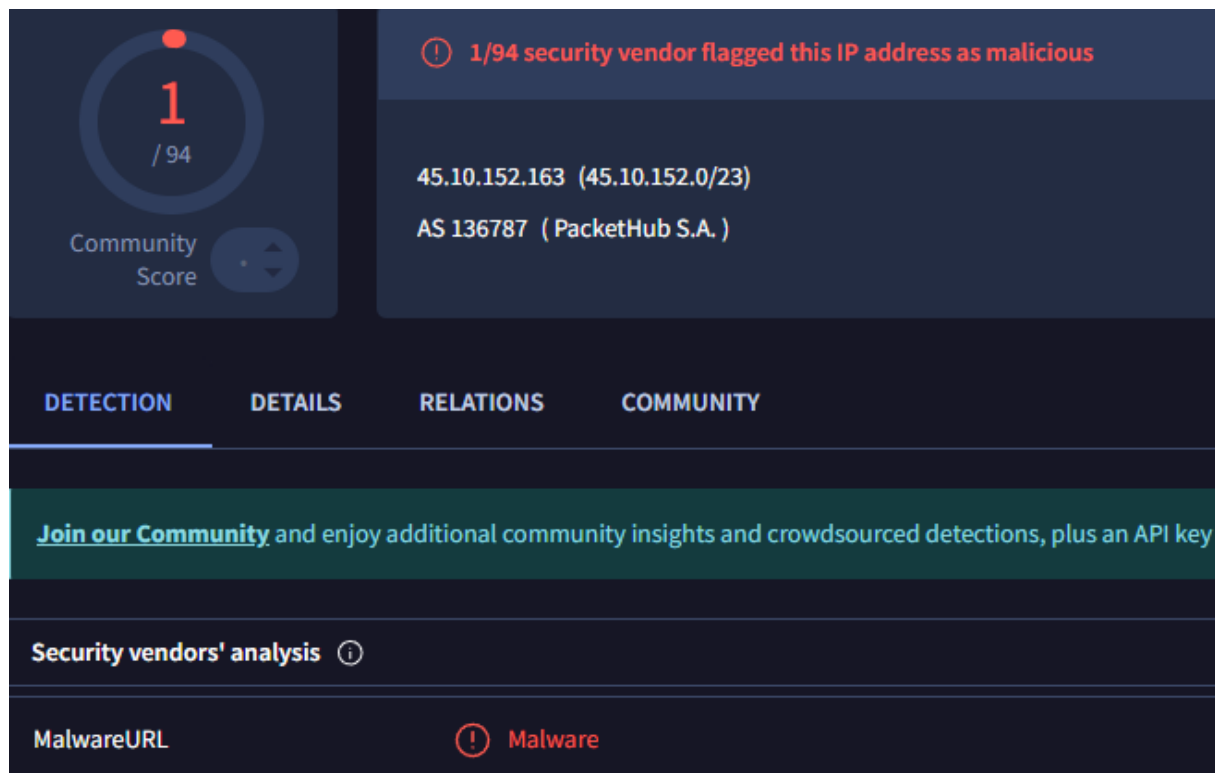
IP: 45.10.152.163 on port 50764 and 51076

***User Agent:* User-Agent mimics Googlebot but lacks full client metadata. Likely indicative of automated probing or scripted attack activity attempting to bypass basic detection.**

ANALYST INVESTIGATION:

Virus Total Result: **[here]<https://www.virustotal.com/gui/ip-address/45.10.152.163>]**

***Security Vendors' Analysis from Virus Total:* 10/94 security vendors flagged this IP address as malicious**



Talos Intelligence:

REPUTATION DETAILS:

Email Reputation: *Neutral*

Web Reputation: *Questionable*

BLOCK LISTS:

Talos Security Intelligence Block List

Spam level = Critical

Talos Result:

[here|https://talosintelligence.com/reputation_center/lookup?search=45.10.152.163]

LOCATION DATA

Copenhagen, Denmark

OWNER DETAILS

IP ADDRESS	45.10.152.163
FWD/REV DNS MATCH	No data
HOSTNAME	-
DOMAIN	-
NETWORK OWNER	net1.gmbh

CONTENT DETAILS

CONTENT CATEGORY No established content categories

Think these category details are incorrect?

Submit Content Categorization Ticket

REPUTATION DETAILS

SENDER IP REPUTATION

Neutral

Submit Sender IP Reputation Ticket

WEB REPUTATION

Questionable

Submit Web Reputation Ticket

EMAIL VOLUME DATA

	LAST DAY	LAST MONTH
EMAIL VOLUME	0.0	0.9
VOLUME CHANGE	0%	
SPAM LEVEL	Critical	

BLOCK LISTS

BL.SPAMCOP.NET	Not Listed
CBL.ABUSEAT.ORG	Not Listed
PBL.SPAMHAUS.ORG	Not Listed
SBL.SPAMHAUS.ORG	Not Listed

TALOS SECURITY INTELLIGENCE BLOCK LIST

ADDED TO THE BLOCK LIST	No
-------------------------	----

ShodanResult: [\[here|https://www.shodan.io/host/45.10.152.163\]](https://www.shodan.io/host/45.10.152.163)

Open Ports: 43194

45.10.152.163

Regular View

Raw Data

Timeline

General Information

Country

Denmark

City

Copenhagen

Organization

PacketHub S.A.

ISP

PacketHub S.A.

ASN

AS136787

Open Ports

43194

// 43194 / UDP

DHT Nodes

103.204.86.53

26463

115.28.162.146

47707

144.243.177.171

33024

159.152.45.203

54041

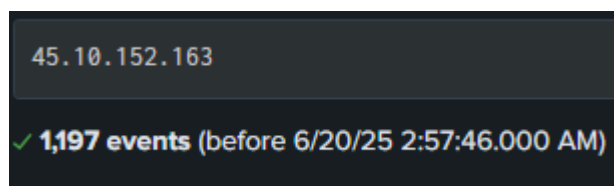
26.224.174.125

50709

189.240.97.39

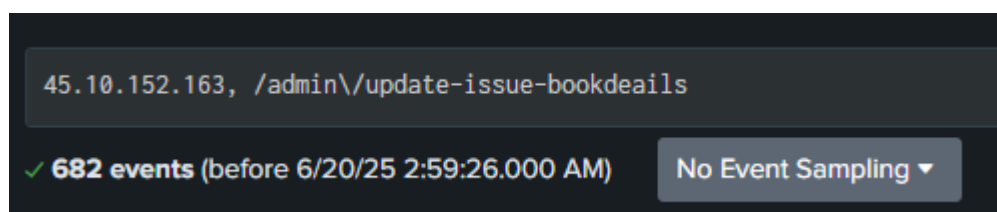
59202

Splunk Investigation:* A total of **1197 logs were found, and after analyzing them with different identifiers, it was determined that **682** logs were associated with the identifier `/admin/update-issue-bookdeails`, and **206** logs were linked to the identifier containing `SLEEP(5)`.*



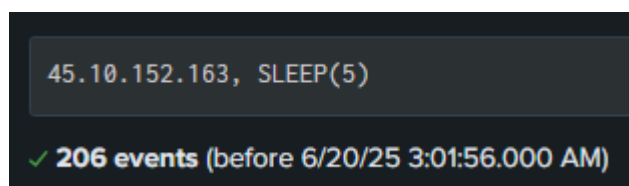
↑ ↑ ↑

Result 1: Splunk search — internal link (not accessible)



↑ ↑ ↑

Result 2: Splunk search — internal link (not accessible)



↑ ↑ ↑

Result 3: Splunk search — internal link (not accessible)

1 Raw Data:


```
{ "timestamp": "2025-04-21T18:09:12.980714-0400", "flow_id": 983666730205291, "in_iface": "eth0", "event_type": "alert", "src_ip": "45.10.152.163", "src_port": 50764, "dest_ip": "[internal web server]", "dest_port": 80, "proto": "TCP", "tx_id": 0, "alert": { "action": "allowed", "gid": 1, "signature_id": 2006446, "rev": 11, "signature": "ET WEB_SERVER Possible SQL Injection Attempt UNION SELECT", "category": "Web Application Attack", "severity": 1 }, "http": { "hostname": "redacted-domain.local", "url": "\admin\update-issue-bookdeails.php?rid=%28SELECT%20%28CASE%20WHEN%20%286210%3D6210%29%20THEN%207%20ELSE%20%28SELECT%208233%20UNION%20SELECT%204070%29%20END%29%29", "http_user_agent": "Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)", "http_content_type": "text/html", "http_method": "GET", "protocol": "HTTP/1.1", "status": 403, "length": 285 }
```

2 Raw Data:

```
{ "timestamp": "2025-04-21T18:09:15.471244-0400", "flow_id": 1861549455820221, "in_iface": "eth0", "event_type": "alert", "src_ip": "45.10.152.163", "src_port": 51076, "dest_ip": "[internal web server]", "dest_port": 80, "proto": "TCP", "http": { "hostname": "redacted-domain.local", "url": "\admin\update-issue-bookdeails.php?rid=7%29%20AND%20%28SELECT%205449%20FROM%20%28SELECT%28SLEEP%285%29%29%29xhtk%29%20AND%20%283162%3D3162", "http_user_agent": "Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)", "http_content_type": "text/html", "http_method": "GET", "protocol": "HTTP/1.1", "status": 403, "length": 285 }, "tx_id": 0, "alert": { "action": "allowed", "gid": 1, "signature_id": 2016935, "rev": 2, "signature": "ET WEB_SERVER SQL Injection Select Sleep Time Delay", "category": "Web Application Attack", "severity": 1 }
```

Additional Findings:

45.10.152.163 was not found in our database

ISP	Packethub S.A.
Usage Type	Data Center/Web Hosting/Transit
ASN	Unknown
Domain Name	packethub.net
Country	 Denmark
City	Copenhagen, Capital Region

IP info including ISP, Usage Type, and Location provided by [IPInfo](#). Updated biweekly.

REPORT 45.10.152.163

WHOIS 45.10.152.163

ResultAbuseIPDB: [[here](https://www.abuseipdb.com/check/45.10.152.163)]

ANALYST ASSESSMENT

The analysis of network traffic logs reveals that the IP address **45.10.152.163** conducted multiple distinct web application attacks targeting the web server at [internal web server] - redacted-domain.local .

Attempt 1: SQL Injection via UNION SELECT

The first attack involved a classic SQL Injection attempt leveraging a **UNION SELECT** statement embedded in the **rid** parameter of the URL **/admin/update-issue-bookdeails.php**. The attacker sent a specially crafted GET request designed to extract data by manipulating the database query logic. This activity was identified by the ET signature "Possible SQL Injection Attempt UNION SELECT" with severity level 1. The

request used a spoofed Googlebot User-Agent, indicating an attempt to bypass detection by masquerading as a legitimate crawler.

Attempt 2: Time-Based Blind SQL Injection Using SLEEP()

The second attack was a time-based blind SQL Injection leveraging the SQL **SLEEP(5)** function in the same vulnerable parameter. This method aims to detect vulnerabilities by causing delays in the database response, thereby confirming successful code injection without direct error messages. This attempt was flagged by the ET signature "SQL Injection Select Sleep Time Delay" with severity level 1. The User-Agent was similarly spoofed as Googlebot.

Both attacks targeted port 80 on the victim server and generated a significant volume of traffic, with **682 logs associated with the /admin/update-issue-bookdeails endpoint** and **206 logs containing the SLEEP(5) payload**. The use of a well-known User-Agent spoofing technique and repeated attempts suggest automated scanning or exploitation tools.

While these attacks are flagged at a moderate severity level (1), the presence of multiple attempts and use of advanced SQL injection techniques (including time delays) indicate a persistent threat actor probing the server for vulnerabilities. The IP address should be considered hostile and subject to blocking and further investigation.

Attack Details:1.

SQL Injection Attempt via UNION SELECT

Request Details:

The attacker from IP address 45.10.152.163 made an HTTP GET request to the following URL:

/admin/update-issue-bookdeails.php?rid=(SELECT (CASE WHEN (6210=6210) THEN 7 ELSE (SELECT 8233 UNION SELECT 4070) END))

Objective:

This attack attempts to exploit SQL Injection vulnerability by injecting a **UNION SELECT** statement. The goal is to retrieve unauthorized data from the database by manipulating the query logic.

Method:

GET – The attacker crafted the URL parameter to inject malicious SQL code.

User-Agent:

A spoofed Googlebot User-Agent was used, likely to evade detection by appearing as legitimate crawler traffic.

Content-Type:

text/html

HTTP Status:

403 Forbidden – The server blocked the request, preventing exploitation.

2. Time-Based Blind SQL Injection Using SLEEP(5)**Request Details:**

The same attacker issued another HTTP GET request targeting the same endpoint with payload:

```
/admin/update-issue-bookdeails.php?rid=7) AND (SELECT 5449  
FROM (SELECT(SLEEP(5)))xhtk) AND (3162=3162
```

Objective:

This is a time-based blind SQL Injection designed to verify vulnerability by causing the database to delay response by 5 seconds using the **SLEEP(5)** function. Such delay indicates successful injection even without direct error messages.

Method:

GET – Used to test and exploit the SQL vulnerability via timing analysis.

User-Agent:

Again, spoofed as Googlebot to avoid detection.

Content-Type:

text/html

HTTP Status:

403 Forbidden – Request was denied by the server.

Goals of these attacks:

1. **/admin/update-issue-bookdeails.php?rid=... (UNION SELECT)**
Goal: Data exfiltration via SQL Injection by manipulating query results.
2. **/admin/update-issue-bookdeails.php?rid=...SLEEP(5)...**
Goal: Confirming SQL Injection vulnerability using time delay to detect successful code injection.

Detection Context:

Suricata Signatures Triggered:

The attacks were detected by the following Suricata Emerging Threats (ET) signatures:

- ET WEB_SERVER Possible SQL Injection Attempt UNION SELECT — indicating a SQL Injection attempt aiming at data extraction.
- ET WEB_SERVER SQL Injection Select Sleep Time Delay — indicating a time-based blind SQL Injection attempt using the **SLEEP()** function.

Unusual URL Access:

The malicious requests targeted the vulnerable endpoint **/admin/update-issue-bookdeails.php** with suspicious SQL payloads in the **rid** parameter:

- The first request used a **UNION SELECT** statement to manipulate query results.
- The second request used a **SLEEP(5)** payload to induce a delay, confirming blind SQL Injection vulnerability.

Suspicious User Agent:

Both requests used a User-Agent string mimicking Googlebot (**Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)**), which lacks full browser metadata. This spoofing is commonly used by automated tools to evade detection.

HTTP Status 403 Forbidden:

The web server returned a 403 Forbidden status for both requests, indicating that the attack attempts were blocked and did not succeed in exploitation.

Traffic Anomalies:

The source IP address 45.10.152.163 has shown repeated malicious activity, including multiple attempts to exploit SQL Injection vulnerabilities, and should be considered hostile based on both IDS detection and threat intelligence context.

ACTION

1. Block the Attacker's IP (45.10.152.163)
Immediately block the IP address 45.10.152.163 on network perimeter

devices and firewalls to prevent further exploitation attempts targeting the vulnerable web application.

2. Harden Web Application Endpoint

Review and secure the `/admin/update-issue-bookdeails.php` endpoint to validate and sanitize all input parameters, especially `rid`, to mitigate SQL injection vulnerabilities.

3. Monitor and Restrict Suspicious User-Agent

Flag or block HTTP requests spoofing the Googlebot User-Agent string `Mozilla/5.0 (compatible; Googlebot/2.1; http://www.google.com/bot.html)`, since it is being abused to bypass detection.

4. Analyze Logs and Investigate Related Activity

Conduct a thorough review of historical logs to identify additional attack attempts from this IP or similar patterns using SQL injection techniques.

5. Update and Patch Systems

Ensure that the web server, database, and related software components are up to date with all security patches applied to reduce attack surface.

6. Implement Web Application Firewall (WAF) Rules

Deploy or tune WAF rules to detect and block SQL injection payloads such as `UNION SELECT` and `SLEEP()` based time delays.

7. Enhance Access Controls and Alerts

Tighten access controls to the application and set up monitoring alerts for suspicious requests targeting critical endpoints.