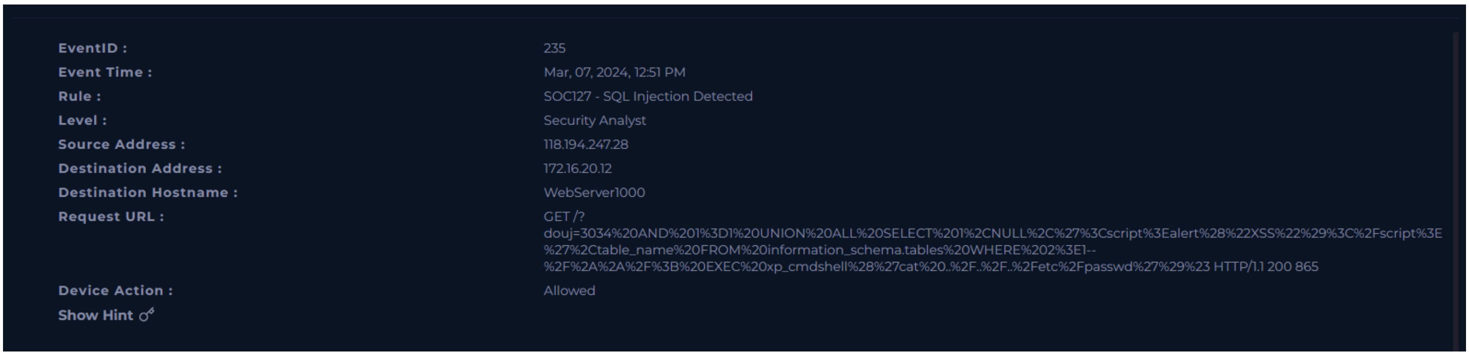
# EventID235 - SOC127 - SQL Injection Detected

## 1. Alert Overview

**Alert Name / Title:** SOC127 – SQL Injection Detected
**Alert Source:** Web Application Firewall / Proxy Logs / SIEM (Log Management Platform)
**Alert Severity:** High
**Detection Rule / Query:** Triggered based on a web request pattern matching SQL Injection signatures and behavioral anomalies in HTTP GET requests targeting vulnerable parameters.
**Date & Time Observed:** Mar, 07, 2024, 12:51 PM



| | |
|---|---|
| EventID : | 235 |
| Event Time : | Mar, 07, 2024, 12:51 PM |
| Rule : | SOC127 - SQL Injection Detected |
| Level : | Security Analyst |
| Source Address : | 118.194.247.28 |
| Destination Address : | 172.16.20.12 |
| Destination Hostname : | WebServer1000 |
| Request URL : | GET /?<br>douj=3034%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%2C%27%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E%27%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--%2F%2A%2A%2F%3B%20EXEC%20xp_cmdshell%28%27cat%20..%2F..%2F..%2Fetc%2Fpasswd%27%29%23 HTTP/1.1 200 865 |
| Device Action : | Allowed |
| Show Hint | |

---

## 2. Initial Alert Details

**Alert Description:**
An alert was triggered due to multiple SQL injection attempts targeting a web application hosted at IP `172.16.20.12`. The malicious traffic originated from the external IP `118.194.247.28` (China).

**Triggered Host / User / Source:**

- **Source IP:** 118.194.247.28
- **Destination Host:** WebServer (172.16.20.12)
- **User:** Anonymous HTTP client (no authentication headers observed)

**Event Count / Frequency:**
Multiple malicious HTTP GET requests detected from the same IP across different ports prior to the SQLi attempt — indicative of reconnaissance activity.

**Detection Context:**
Behavioral and signature-based detection from proxy/firewall logs. The alert fired due to the presence of SQL keywords (`UNION ALL SELECT`, `xp_cmdshell`, `EXTRACTVALUE`, etc.) in HTTP parameters.

---

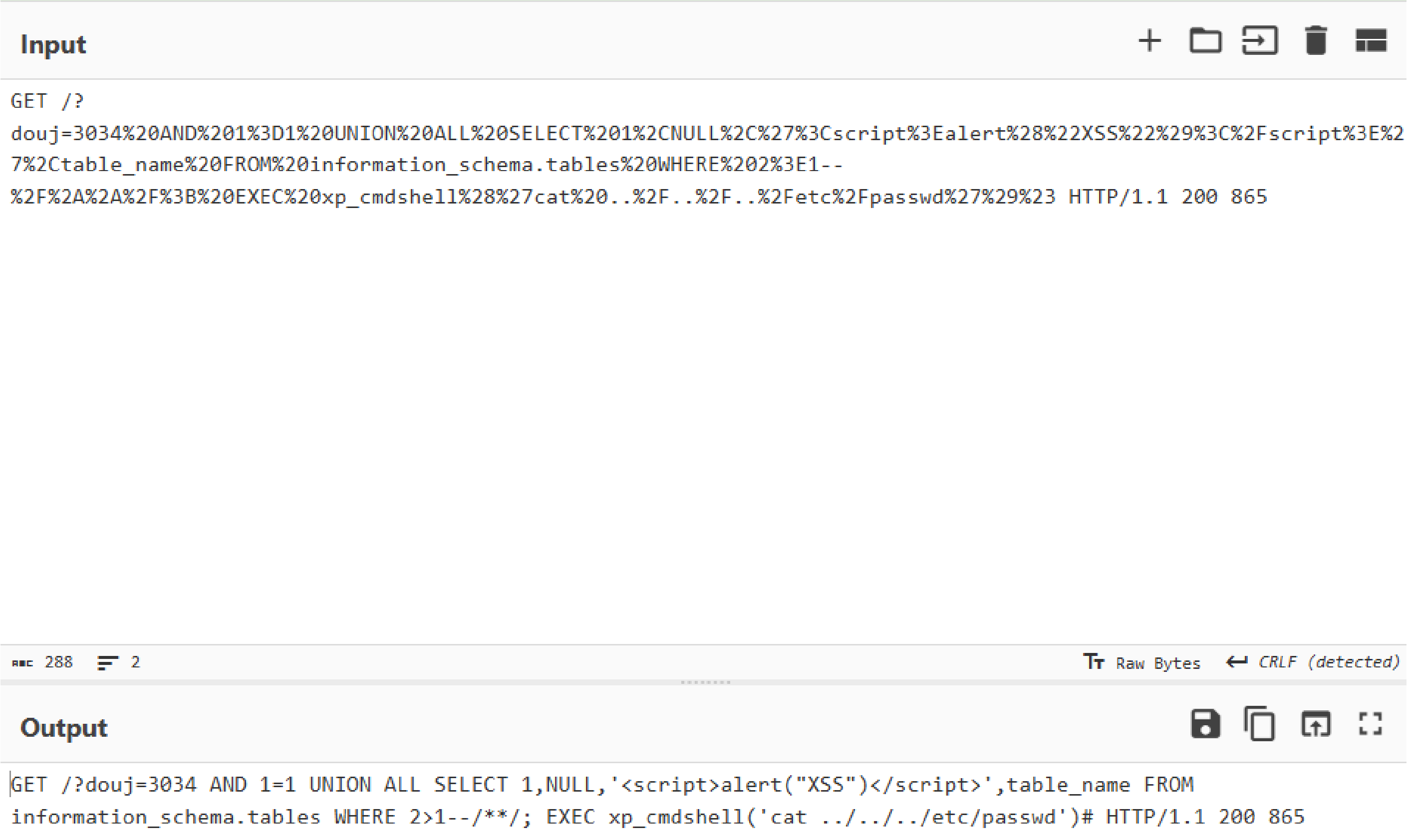
## 3. Investigation Steps

**Step 1: Verification of the Alert**

- Queried log management for all requests originating from `118.194.247.28`.
- Observed several malicious requests containing SQL keywords, encoded payloads, and suspicious parameters.
- A representative request:

  ```
  GET /?douj=3034 AND 1=1 UNION ALL SELECT 1,NULL,'<script>alert("XSS")
  </script>',table_name FROM information_schema.tables WHERE 2>1--/**/; EXEC
  xp_cmdshell('cat ../../../../etc/passwd')#
  ```

- Result: HTTP `200 OK` response observed, confirming the request reached the server successfully.

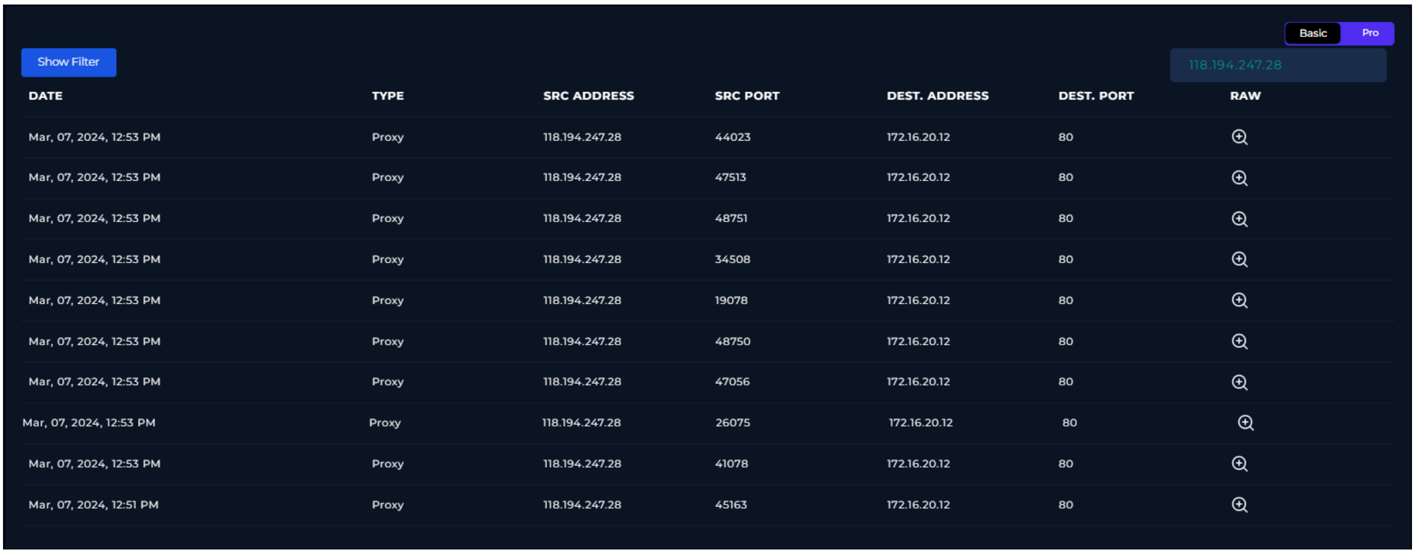


```
GET /?
douj=3034%20AND%201%3D1%20UNION%20ALL%20SELECT%201%2CNULL%2C%27%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E%2
7%2Ctable_name%20FROM%20information_schema.tables%20WHERE%202%3E1--
%2F%2A%2A%2F%3B%20EXEC%20xp_cmdshell%28%27cat%20..%2F..%2F..%2Fetc%2Fpasswd%27%29%23 HTTP/1.1 200 865
```

```
GET /?douj=3034 AND 1=1 UNION ALL SELECT 1,NULL,'<script>alert("XSS")</script>',table_name FROM
information_schema.tables WHERE 2>1--/**/; EXEC xp_cmdshell('cat ../../../etc/passwd')# HTTP/1.1 200 865
```

### Step 2: Cross-check with Other Data Sources

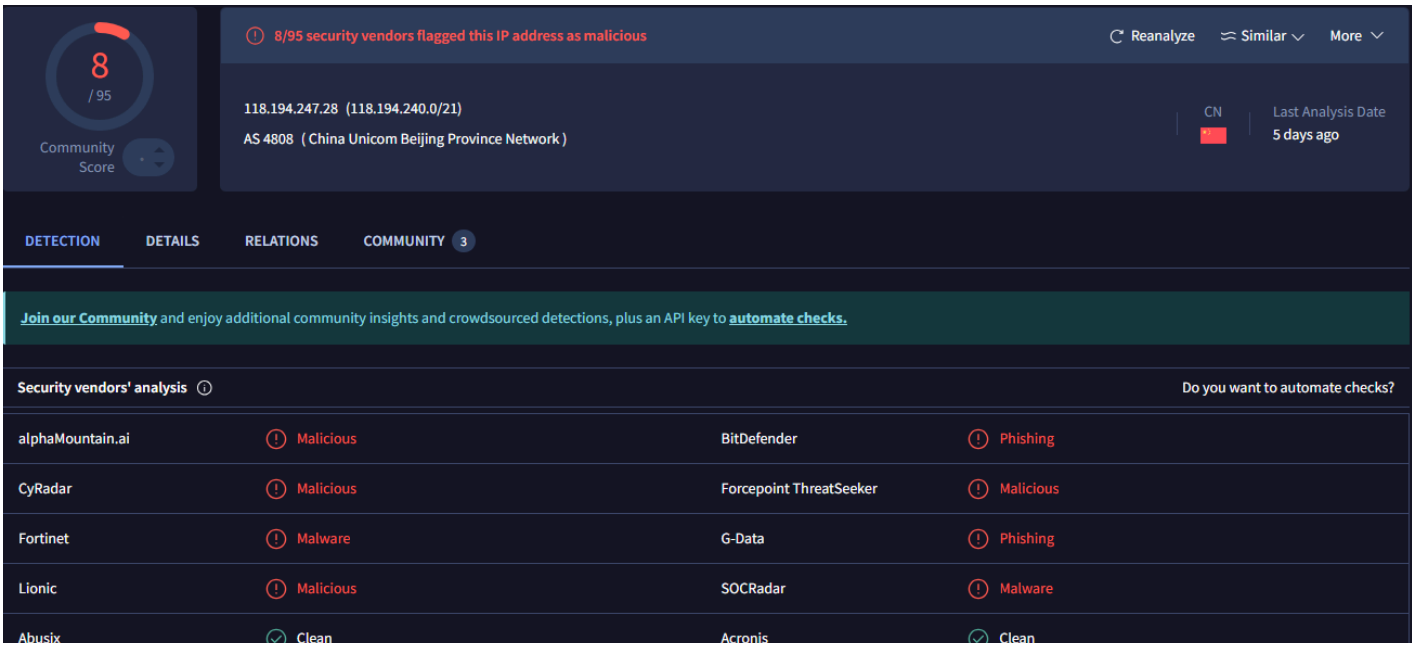
- Proxy and firewall logs showed previous port scan activity from the same IP, specifically to port 80.
- This indicates the attacker first conducted reconnaissance before exploiting the public-facing web server.
- Timeline analysis suggests sequential scanning → exploitation attempt.



| DATE | TYPE | SRC ADDRESS | SRC PORT | DEST. ADDRESS | DEST. PORT | RAW |
|---|---|---|---|---|---|---|
| Mar, 07, 2024, 12:53 PM | Proxy | 118.194.247.28 | 44023 | 172.16.20.12 | 80 | ⊕ |
| Mar, 07, 2024, 12:53 PM | Proxy | 118.194.247.28 | 47513 | 172.16.20.12 | 80 | ⊕ |
| Mar, 07, 2024, 12:53 PM | Proxy | 118.194.247.28 | 48751 | 172.16.20.12 | 80 | ⊕ |
| Mar, 07, 2024, 12:53 PM | Proxy | 118.194.247.28 | 34508 | 172.16.20.12 | 80 | ⊕ |
| Mar, 07, 2024, 12:53 PM | Proxy | 118.194.247.28 | 19078 | 172.16.20.12 | 80 | ⊕ |
| Mar, 07, 2024, 12:53 PM | Proxy | 118.194.247.28 | 48750 | 172.16.20.12 | 80 | ⊕ |
| Mar, 07, 2024, 12:53 PM | Proxy | 118.194.247.28 | 47056 | 172.16.20.12 | 80 | ⊕ |
| Mar, 07, 2024, 12:53 PM | Proxy | 118.194.247.28 | 26075 | 172.16.20.12 | 80 | ⊕ |
| Mar, 07, 2024, 12:53 PM | Proxy | 118.194.247.28 | 41078 | 172.16.20.12 | 80 | ⊕ |
| Mar, 07, 2024, 12:51 PM | Proxy | 118.194.247.28 | 45163 | 172.16.20.12 | 80 | ⊕ |

### Step 3: IP Reputation Analysis

- Checked the IP `118.194.247.28` on **AbuseIPDB** and **VirusTotal**.
- Both sources flagged it as **malicious** for web attacks, phishing, and brute-force attempts.
- Confidence in malicious intent: **High**.



8 / 95

Community Score

8/95 security vendors flagged this IP address as malicious

118.194.247.28 (118.194.240.0/21)

AS 4808 ( China Unicom Beijing Province Network )

CN

Last Analysis Date

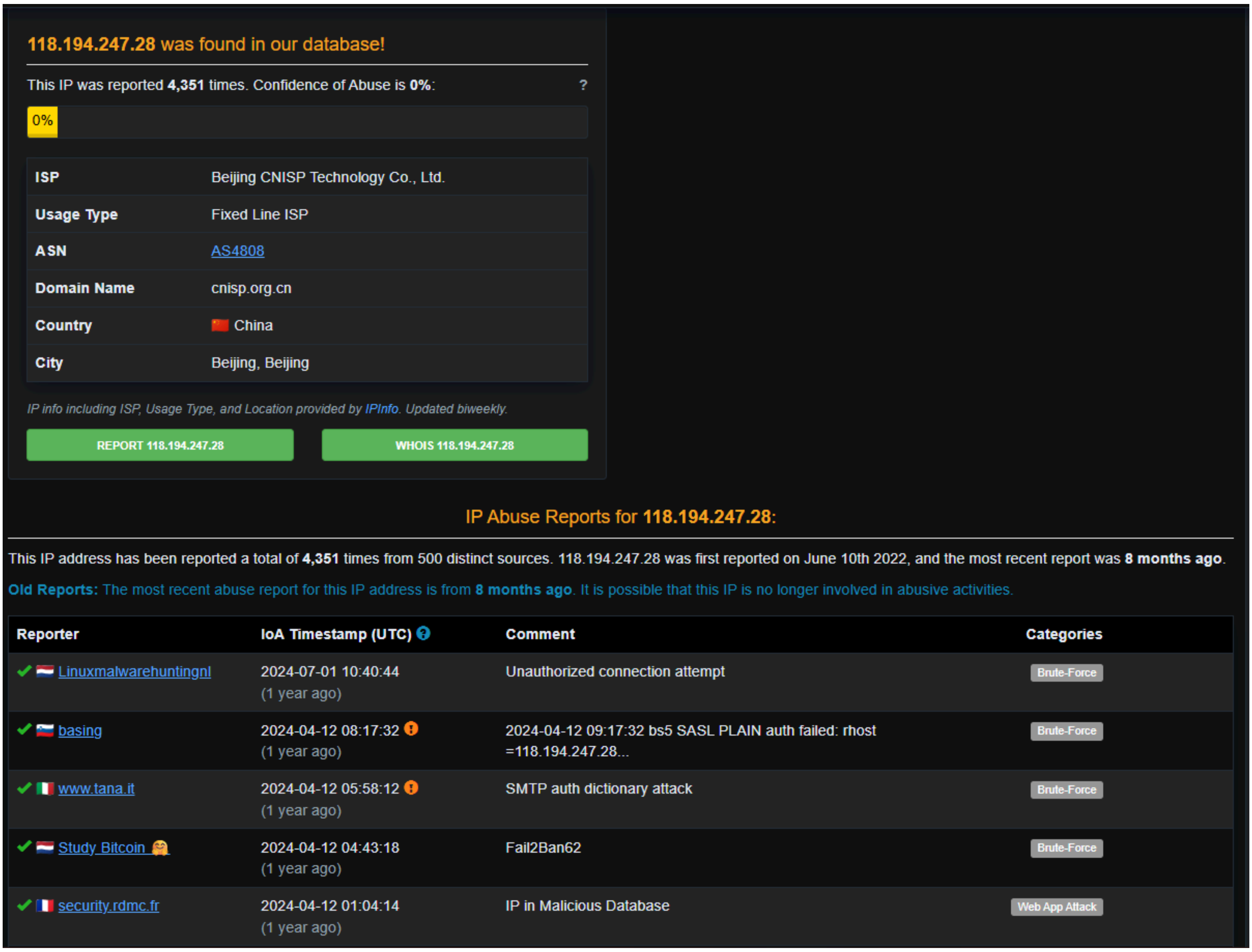
5 days ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Security vendors' analysis

Do you want to automate checks?

| | | | |
|---|---|---|---|
| alphaMountain.ai | Malicious | BitDefender | Phishing |
| CyRadar | Malicious | Forcepoint ThreatSeeker | Malicious |
| Fortinet | Malware | G-Data | Phishing |
| Lionic | Malicious | SOCRadar | Malware |
| Abusix | Clean | Acronis | Clean |



118.194.247.28 was found in our database!

This IP was reported **4,351** times. Confidence of Abuse is **0%**: ?

0%

| | |
|---|---|
| **ISP** | Beijing CNISP Technology Co., Ltd. |
| **Usage Type** | Fixed Line ISP |
| **ASN** | AS4808 |
| **Domain Name** | cnisp.org.cn |
| **Country** | China |
| **City** | Beijing, Beijing |

*IP info including ISP, Usage Type, and Location provided by IPInfo. Updated biweekly.*

IP Abuse Reports for **118.194.247.28**:

This IP address has been reported a total of **4,351** times from 500 distinct sources. 118.194.247.28 was first reported on June 10th 2022, and the most recent report was **8 months ago**.

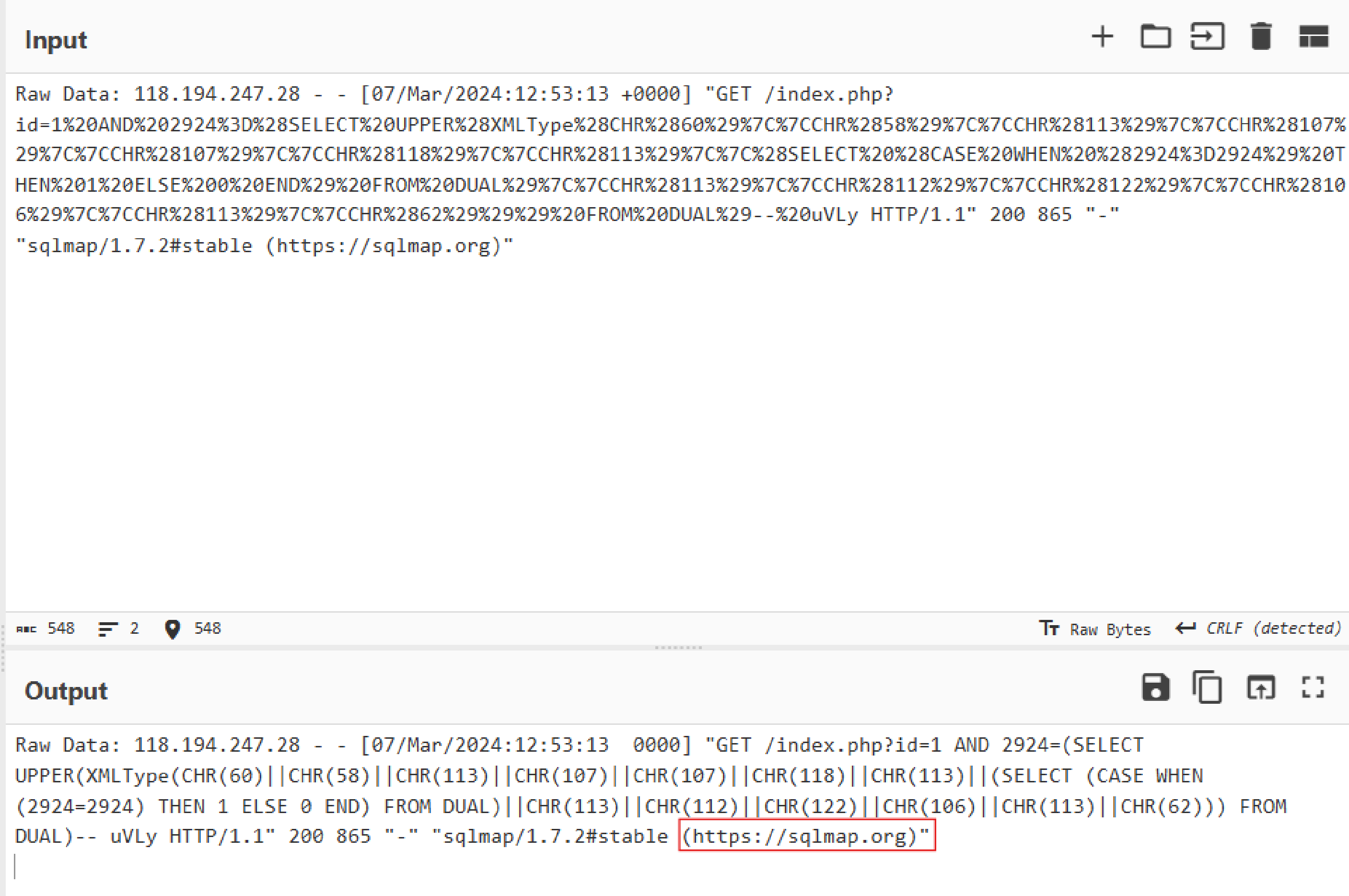
**Old Reports:** The most recent abuse report for this IP address is from **8 months ago**. It is possible that this IP is no longer involved in abusive activities.

| Reporter | IoA Timestamp (UTC) | Comment | Categories |
|---|---|---|---|
| ✓ Linuxmalwarehuntingnl | 2024-07-01 10:40:44<br>(1 year ago) | Unauthorized connection attempt | Brute-Force |
| ✓ basing | 2024-04-12 08:17:32<br>(1 year ago) | 2024-04-12 09:17:32 bs5 SASL PLAIN auth failed: rhost =118.194.247.28... | Brute-Force |
| ✓ www.tana.it | 2024-04-12 05:58:12<br>(1 year ago) | SMTP auth dictionary attack | Brute-Force |
| ✓ Study Bitcoin 🤗 | 2024-04-12 04:43:18<br>(1 year ago) | Fail2Ban62 | Brute-Force |
| ✓ security.rdmc.fr | 2024-04-12 01:04:14<br>(1 year ago) | IP in Malicious Database | Web App Attack |

### Step 4: Payload Analysis & Decoding

- Decoded obfuscated requests using **CyberChef** to reveal structured SQL injection attempts. Examples:
  - `EXTRACTVALUE(7321, CONCAT(...))` → classic MySQL blind SQLi test.

    - `xp_cmdshell('cat ../../../../etc/passwd')` → command execution attempt on the backend OS.
- Confirmed use of **sqlmap** (automated SQL injection tool) based on query patterns and payloads.



```
Raw Data: 118.194.247.28 - - [07/Mar/2024:12:53:13 +0000] "GET /index.php?
id=1%20AND%202924%3D%28SELECT%20UPPER%28XMLType%28CHR%2860%29%7C%7CCHR%2858%29%7C%7CCHR%28113%29%7C%7CCHR%28107%
29%7C%7CCHR%28107%29%7C%7CCHR%28118%29%7C%7CCHR%28113%29%7C%7C%28SELECT%20%28CASE%20WHEN%20%282924%3D2924%29%20T
HEN%201%20ELSE%200%20END%29%20FROM%20DUAL%29%7C%7CCHR%28113%29%7C%7CCHR%28112%29%7C%7CCHR%28122%29%7C%7CCHR%2810
6%29%7C%7CCHR%28113%29%7C%7CCHR%2862%29%29%29%20FROM%20DUAL%29--%20uVLy HTTP/1.1" 200 865 "-"
"sqlmap/1.7.2#stable (https://sqlmap.org)"
```

```
Raw Data: 118.194.247.28 - - [07/Mar/2024:12:53:13  0000] "GET /index.php?id=1 AND 2924=(SELECT
UPPER(XMLType(CHR(60)||CHR(58)||CHR(113)||CHR(107)||CHR(107)||CHR(118)||CHR(113)||(SELECT (CASE WHEN
(2924=2924) THEN 1 ELSE 0 END) FROM DUAL)||CHR(113)||CHR(112)||CHR(122)||CHR(106)||CHR(113)||CHR(62))) FROM
DUAL)-- uVLy HTTP/1.1" 200 865 "-" "sqlmap/1.7.2#stable (https://sqlmap.org)"
```

### Step 5: Correlation & Scope Assessment

- No internal hosts exhibited lateral movement from `172.16.20.12`.
- Database logs did not show any successful data extraction or modification queries.
- Indicates the SQL injection **attempts failed** to execute successfully.

### Step 6: MITRE ATT&CK Mapping

- Reconnaissance → **T1595.002 – Active Scanning: Vulnerability Scanning**
- Initial Access → **T1190 – Exploit Public-Facing Application**
- Credential Access (potential goal) → **T1552.001 – Unsecured Credentials in Files**

---

## 4. Findings

### Summary of Evidence:

- Multiple SQLi payloads with UNION SELECT, CASE WHEN, EXTRACTVALUE, and xp_cmdshell commands.
- Source IP confirmed as malicious via OSINT.
- HTTP 200 responses confirmed the requests were processed but not necessarily exploited.

**Root Cause / Attack Vector:**
Attacker attempted SQL injection against vulnerable web parameters exposed via the public-facing web application ( `index.php` , `douj` , `id` parameters).

**Affected Systems / Users:**

- WebServer: 172.16.20.12
- No evidence of compromised internal users or database accounts.

---

## 5. Analysis Conclusion

**Alert Status: True Positive** (Confirmed SQL Injection attempts)

**Impact Assessment:**

- **Current Impact:** Low – no successful exploitation observed.
- **Potential Impact:** High – could lead to database compromise, command execution, or data exfiltration if vulnerable.

---

## 6. Response & Remediation

**Immediate Actions Taken:**

- Blocked attacker IP ( `118.194.247.28` ) on perimeter firewall.
- Reviewed and hardened web server configurations.
- Validated no evidence of exploitation in backend database logs.

**Recommended Next Steps:**

- Conduct a full vulnerability assessment on the web application.
- Validate input sanitization for all parameters ( `id` , `douj` , etc.).
- Monitor for further malicious traffic using updated WAF rules.

**Preventive Measures:**

- Implement **Web Application Firewall (WAF)** with updated SQLi signatures.
- Enforce **input validation and parameterized queries** in web applications.
- Deploy **Multi-Factor Authentication (MFA)** for administrative interfaces.
- Regularly patch and update web servers and dependent components.

---

## 7. Learning & Improvement

**Lessons Learned:**

- Public-facing applications should not expose remote access unnecessarily.
- Logs indicate early reconnaissance that could have been detected sooner with proper alert thresholds.

**Detection Rule Enhancement:**
Previous rule relied solely on signature matching. Enhanced detection rule should include:

- Rate-based thresholds (multiple SQLi payloads from same IP).
- Correlation with prior port scan activity.

**Knowledge Gained:**

- Improved understanding of SQL injection payload structure and encoding methods.
- Enhanced correlation between reconnaissance and exploitation phases.

---