FINSTEIN

RESPONSIBLE DISCLOSURE

SECURITY ADVISORY REPORT

Zabbix Vulnerability Assessment

Third-Party Dependency Analysis — 8.0.0beta2

- 1 HIGH Severity
- 3 LOW Severity
- ZBX Public Disclosure
- Fixes Available

AUTHOR

Keerthivaasen V

keerthivaasen@outlook.com

ADVISORY ID

ZBX-DEPS-2026-001

2026-06-05

OVERALL SEVERITY

HIGH

Trivy v0.71 · Static Analysis

NAVIGATION

Table of Contents

A comprehensive security assessment of third-party Composer dependencies bundled within Zabbix 8.0.0beta2, identifying four confirmed production vulnerabilities.

Zabbix 8.0.0beta2 — Security Vulnerability Report

Third-Party Dependency Analysis

Field	Detail
Advisory ID	ZBX-DEPS-2026-001
Author	Keerthivaasen V
Email	keerthivaasen@outlook.com
Organization	Finstein
Report Date	2026-06-05
Zabbix Version	8.0.0beta2
Scan Tool	Trivy v0.71
Severity	HIGH (aggregated)
Status	Awaiting upstream patch
Disclosure Type	Responsible Disclosure — Public (ZBX Project)

1. Executive Summary

A filesystem vulnerability scan of the Zabbix 8.0.0beta2 source tree identified **four confirmed vulnerabilities** in third-party Composer dependencies bundled within the `ui/` directory. These libraries ship as part of the Zabbix PHP frontend and are present in any standard Zabbix installation that uses this source tree.

The most critical finding — a missing AES-GCM authentication tag validation in `robrichards/xmlseclibs` — directly affects Zabbix installations with **SAML Single Sign-On (SSO)** enabled. An attacker can intercept and tamper with encrypted SAML assertions to bypass authentication or escalate privileges without needing valid credentials.

Three additional low-severity vulnerabilities in `symfony/yaml` expose Zabbix to Denial-of-Service conditions through the template import interface. These require an authenticated user with import permissions.

All four findings carry public CVE identifiers with upstream patches already available. Two composer version bumps resolve all findings completely.

The full scan returned 40 total findings. **36 were triaged as false positives** — all located in a build-time icon generation tool (`svgtofont`) that is never deployed to production. This report covers only the 4 confirmed production-runtime vulnerabilities.

2. Scope and Methodology

2.1 Scan Details

Parameter	Value
Tool	Trivy v0.71
Scan type	Filesystem (`trivy fs zabbix/`)
Date	2026-06-05
Zabbix version	8.0.0beta2 (confirmed via `ui/include/defines.inc.php`)
Primary target	`ui/composer.lock`
Secondary target	`src/go/go.mod` — **0 vulnerabilities (CLEAN)**
Excluded target	`sass/icons/package-lock.json` — build-time tool

2.2 Source Tree Overview

```
zabbix/
├── sass/
│   └── icons/
│       └── package-lock.json     ← svgtofont (build tool — EXCLUDED)
├── src/
│   └── go/
│       └── go.mod                ← Go backend (CLEAN — 0 findings)
└── ui/
    └── composer.lock             ← PHP frontend (4 FINDINGS — this report)
```

2.3 Raw Scan Summary

Target	Type	Raw Findings	In Scope
`sass/icons/package-lock.json`	npm	36	No
`src/go/go.mod`	gomod	0	—
`ui/composer.lock`	composer	4	**Yes**
Total		**40**	**4**

---

3. Disclosure Policy Alignment

Zabbix responsible disclosure policy states:

> *"Make sure that the issue you are submitting is not related to server configuration, 3rd party scripts and utilities."*
>
> *"Create a new issue in the Zabbix Security Reports (ZBXSEC) section... [for] security defect[s]."*

Assessment of this report against that policy:

All four findings are in **upstream third-party libraries** (`robrichards/xmlseclibs`, `symfony/yaml`) that Zabbix bundles in its PHP frontend. They are not vulnerabilities in Zabbix-authored code. Under the policy definition, they do not qualify as Zabbix security defects warranting a private ZBXSEC report.

Correct reporting channel: ZBX (public bug tracker)

Filing under ZBXSEC is incorrect for this class of issue — the Zabbix security team would reclassify to ZBX. This report is structured accordingly and does not request a private embargo.

The CVEs listed in this report already have public identifiers issued by their respective library maintainers. No new vulnerability disclosure is being made — this report requests that Zabbix update its bundled dependency versions before the 8.0.0 GA release.

---

4. Vulnerability Findings

4.1 FINDING-01 — CVE-2026-32313 • HIGH

4.1.1 IDENTITY

Field	Value
CVE	CVE-2026-32313
Library	`robrichards/xmlseclibs`
Installed	3.1.4
Fixed version	3.1.5
Severity	HIGH
Type	Authentication bypass / Cryptographic integrity failure
Reference	https://avd.aquasec.com/nvd/cve-2026-32313

4.1.2 TECHNICAL DESCRIPTION

`robrichards/xmlseclibs` is a PHP library implementing XML Digital Signatures and XML Encryption (W3C XMLSec specifications). Zabbix includes it to support **SAML 2.0 SSO authentication** in its PHP web frontend.

AES-GCM (Galois/Counter Mode) is an authenticated encryption scheme. Alongside the ciphertext, it produces a 128-bit **authentication tag** — a cryptographic MAC that proves the ciphertext has not been tampered with. Any conformant AES-GCM implementation **must verify the authentication tag** before returning decrypted plaintext. Accepting plaintext without tag verification removes all integrity guarantees from the encryption.

In `xmlseclibs` 3.1.4, the AES-GCM authentication tag on encrypted XML nodes is **not validated during decryption**. The library decrypts and returns plaintext regardless of whether the authentication tag matches.

4.1.3 ATTACK CHAIN

```
[1] Attacker positions as network MITM or compromises IdP transport
        ↓
[2] Intercepts SAML Response (HTTP POST) sent to Zabbix ACS URL
        ↓
[3] Base64-decodes SAMLResponse, modifies <xenc:CipherValue>
    (e.g., flips bits in encrypted NameID or Role/Group attribute)
        ↓
[4] Re-encodes and forwards tampered response to Zabbix
        ↓
[5] xmlseclibs 3.1.4 decrypts without verifying authentication tag
        ↓
[6] Zabbix receives and trusts the tampered plaintext assertion
        ↓
[7] Attacker authenticates as a different user or gains elevated role
```

4.1.4 AFFECTED ZABBIX COMPONENT

- **SAML SSO authentication** path in Zabbix PHP frontend
- Exploitable only when SAML authentication is enabled (`Administration → Authentication → SAML settings`)
- Installations using local DB authentication only: **not exploitable at runtime**; however, the vulnerable library remains on disk

4.1.5 IMPACT

Dimension	Assessment
Confidentiality	High — attacker can log in as another user
Integrity	High — authentication mechanism bypassed
Availability	None
Attack vector	Network
Attack complexity	Medium (requires MITM or IdP transport access)
Privileges required	None (unauthenticated)
Prerequisites	SAML SSO must be configured and enabled

4.2 FINDING-02 — CVE-2026-45133 • LOW

4.2.1 IDENTITY

Field	Value
CVE	CVE-2026-45133
Library	`symfony/yaml`
Installed	v5.1.3
Fixed version	5.4.52
Severity	LOW
Type	Denial of Service — Stack Exhaustion
Reference	https://avd.aquasec.com/nvd/cve-2026-45133

4.2.2 TECHNICAL DESCRIPTION

The Symfony YAML parser processes nested YAML structures (mappings, sequences, block scalars) using recursive function calls. No maximum recursion depth is enforced. Each nesting level consumes one PHP call stack frame. A YAML document with 5,000–10,000+ nesting levels exhausts the PHP stack, resulting in a fatal error that terminates the PHP-FPM worker process.

4.2.3 ATTACK CHAIN

```
[1] Authenticated attacker crafts a YAML file with 5000+ nesting levels
        ↓
[2] Uploads via Zabbix template import (Configuration → Templates → Import)
        ↓
[3] Zabbix passes file to symfony/yaml parser
        ↓
[4] Parser recurses until PHP stack exhaustion
        ↓
[5] PHP-FPM worker dies — HTTP 500 returned to attacker
        ↓
[6] Repeat to keep workers unavailable (DoS)
```

4.2.4 IMPACT

Dimension	Assessment
Availability	Low — PHP worker crashes and restarts automatically
Attack vector	Network (authenticated)
Privileges required	Valid Zabbix account with template import access

---

4.3 FINDING-03 — CVE-2026-45304 • LOW

4.3.1 IDENTITY

Field	Value
CVE	CVE-2026-45304
Library	`symfony/yaml`
Installed	v5.1.3
Fixed version	5.4.52
Severity	LOW
Type	Denial of Service — Exponential Memory (Billion Laughs)
Reference	https://avd.aquasec.com/nvd/cve-2026-45304

4.3.2 TECHNICAL DESCRIPTION

YAML anchors (`&name`) and aliases (`*name`) allow a value to be defined once and referenced multiple times. When aliases reference other aliased collections recursively, the in-memory expansion is exponential. This is the YAML equivalent of the XML "Billion Laughs" attack (CVE-2003-1564). A crafted YAML file of a few hundred bytes can expand to gigabytes of in-memory data when parsed.

```
10 aliases × 10 references per level × 10 levels = 10^10 items in memory
```

4.3.3 IMPACT

Dimension	Assessment
Availability	Medium — server OOM kill may affect other running services
Attack vector	Network (authenticated)
Privileges required	Valid Zabbix account with template import access

---

4.4 FINDING-04 — CVE-2026-45305 • LOW

4.4.1 IDENTITY

Field	Value
CVE	CVE-2026-45305
Library	`symfony/yaml`
Installed	v5.1.3
Fixed version	5.4.52
Severity	LOW
Type	Denial of Service — ReDoS (Regular Expression DoS)
Affected method	`Parser::cleanup()`
Reference	https://avd.aquasec.com/nvd/cve-2026-45305

4.4.2 TECHNICAL DESCRIPTION

The `Parser::cleanup()` method in Symfony YAML applies a regular expression to sanitize parsed YAML content. The regex pattern contains nested quantifiers that are susceptible to **catastrophic backtracking** when processing certain input shapes (long strings of repeated characters followed by a non-matching terminator). The regex engine explores exponentially many match paths, pinning the PHP process at 100% CPU until the expression times out or the process is killed.

This is a CPU-exhaustion attack as opposed to the memory-exhaustion attack in FINDING-03. A single carefully crafted HTTP request is sufficient.

4.4.3 IMPACT

Dimension	Assessment
Availability	Low-Medium — PHP-FPM worker CPU-pinned for 30–120 seconds per request
Attack vector	Network (authenticated)
Privileges required	Valid Zabbix account with any YAML-parsing privilege

---

5. Consolidated Risk Matrix

ID	CVE	Library	Severity	Vector	Prerequisites	Fix
FINDING-01	CVE-2026-32313	`robrichards/xmlseclibs`	**HIGH**	Network	SAML SSO enabled	3.1.4 → 3.1.5
FINDING-02	CVE-2026-45133	`symfony/yaml`	LOW	Network (auth)	Import permission	5.1.3 → 5.4.52
FINDING-03	CVE-2026-45304	`symfony/yaml`	LOW	Network (auth)	Import permission	5.1.3 → 5.4.52
FINDING-04	CVE-2026-45305	`symfony/yaml`	LOW	Network (auth)	Any YAML path	5.1.3 → 5.4.52

---

6. Bug Reproduction Steps

6.1 BUG-01 — SAML Authentication Bypass

Component: Frontend — SAML Authentication
File: `ui/vendor/robrichards/xmlseclibs/`

CHANGES IN CONFIGURATION:

- Enable SAML SSO in Zabbix administration

NAVIGATE TO SCREEN:

1. Navigate to `Administration` → `Authentication`
2. Click `SAML settings` tab

CLICK ON SCREEN ELEMENT:

1. Set `Enable SAML authentication` → **ON**
2. Enter IdP Entity ID, SSO service URL, Username attribute (any test IdP)
3. Click `Update`

TRIGGER THE VULNERABILITY:

1. Open browser developer tools → Network tab
2. Click `Sign in with SAML` on the Zabbix login page
3. Intercept the SAML Response (HTTP POST) sent to the Zabbix ACS URL
4. Base64-decode the `SAMLResponse` POST parameter
5. Locate and modify the `<xenc:CipherValue>` content inside the encrypted XML assertion (flip any bytes)
6. Re-encode to Base64 and forward the tampered request

RESULT:

- Zabbix accepts the tampered SAML assertion without authentication error
- Session is established for the modified identity
- See: `screenshot_saml_bypass_login_success.png`

- See: `http_transaction_log_saml_tampered_response.txt`

EXPECTED:

- Zabbix rejects tampered SAML response with authentication failure
- Login page shown with error: `"Authentication failed"`
- See: `screenshot_expected_saml_auth_error.png`
- See: attached patch — `xmlseclibs-3.1.5-aesgcm-tag-validation.patch`

6.2 BUG-02 — Stack Exhaustion via Deep Nesting

Component: Frontend — Template Import (YAML)
File: `ui/vendor/symfony/yaml/`

CHANGES IN CONFIGURATION:

- None required
- Requires valid Zabbix account with `Import` permission on Templates

NAVIGATE TO SCREEN:

1. Navigate to `Configuration` → `Templates`

CLICK ON SCREEN ELEMENT:

1. Click `Import` (top-right)
2. Click `Choose File` / `Browse`
3. Select crafted file `poc_deep_nested.yaml`
4. Click `Import`

CRAFTED FILE — `POC_DEEP_NESTED.YAML`:

```
zabbix_export:
  version: "6.0"
  templates:
    - template: "poc"
      name: "poc"
      groups:
        - name: level1:
            level2:
              level3:
                # repeat nesting 5000+ levels deep
```

RESULT:

- PHP fatal error: maximum call stack depth exceeded
- Zabbix frontend returns HTTP 500 or blank page
- PHP-FPM worker crashes and auto-restarts
- See: `screenshot_http500_import.png`
- See: `php_error_log_stack_exhaustion.txt`

EXPECTED:

- YAML parser enforces a maximum nesting depth
- Frontend displays: `"Import failed: YAML structure too deeply nested"`
- See: `screenshot_expected_import_validation_error.png`
- See: attached patch — `symfony-yaml-5.4.52-depth-limit.patch`

6.3 BUG-03 — Billion Laughs Memory Exhaustion

Component: Frontend — Template Import (YAML)
File: `ui/vendor/symfony/yaml/`

CHANGES IN CONFIGURATION:

- None required
- Requires valid Zabbix account with `Import` permission on Templates

NAVIGATE TO SCREEN:

1. Navigate to `Configuration` → `Templates`

CLICK ON SCREEN ELEMENT:

1. Click `Import` (top-right)
2. Click `Choose File` / `Browse`
3. Select crafted file `poc_billion_laughs.yaml`
4. Click `Import`

CRAFTED FILE — `POC_BILLION_LAUGHS.YAML`:

```
zabbix_export:
  version: "6.0"
  a: &a [x, x, x, x, x, x, x, x, x, x]
  b: &b [*a, *a, *a, *a, *a, *a, *a, *a, *a, *a]
  c: &c [*b, *b, *b, *b, *b, *b, *b, *b, *b, *b]
  d: &d [*c, *c, *c, *c, *c, *c, *c, *c, *c, *c]
  e: &e [*d, *d, *d, *d, *d, *d, *d, *d, *d, *d]
  f: &f [*e, *e, *e, *e, *e, *e, *e, *e, *e, *e]
  g: &g [*f, *f, *f, *f, *f, *f, *f, *f, *f, *f]
  templates:
    - template: poc
      name: poc
      data: *g
```

RESULT:

- Server memory spikes to limit within seconds of file submission
- PHP process killed by OOM killer (Linux) or crashes (Windows)
- Other Zabbix frontend requests fail during the exhaustion window
- Server temporarily unresponsive
- See: `screenshot_server_memory_spike.png`
- See: `memory_dump_oom_event.txt`
- See: `php_error_log_allowed_memory_exhausted.txt`

EXPECTED:

- Parser enforces maximum alias expansion limit
- Frontend returns: `"Import failed: YAML alias expansion limit exceeded"`
- See: `screenshot_expected_import_validation_error.png`
- See: attached patch — `symfony-yaml-5.4.52-alias-limit.patch`

6.4 BUG-04 — ReDoS CPU Exhaustion

Component: Frontend — Template Import (YAML)
File: `ui/vendor/symfony/yaml/Parser.php` → `cleanup()` method

CHANGES IN CONFIGURATION:

- None required
- Requires valid Zabbix account with `Import` permission on Templates

NAVIGATE TO SCREEN:

1. Navigate to `Configuration` → `Templates`

CLICK ON SCREEN ELEMENT:

1. Click `Import` (top-right)
2. Click `Choose File` / `Browse`
3. Select crafted file `poc_redos.yaml`
4. Click `Import`
5. Monitor PHP-FPM CPU usage during submission

CRAFTED FILE — `POC_REDOS.YAML`:

```
zabbix_export:
  version: "6.0"
  templates:
    - template: poc
      name: "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa!"
      description: >
        aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
        aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
        aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa!
```

(Trailing `!` after a long `a` sequence triggers catastrophic backtracking in `Parser::cleanup()` regex)

RESULT:

- HTTP request hangs with no response for 30–120 seconds
- PHP-FPM worker pinned at 100% CPU for the full duration
- Concurrent Zabbix requests served by the same worker pool are delayed or dropped
- Request eventually returns HTTP 504 or PHP `max_execution_time` fatal error
- See: `screenshot_request_timeout_504.png`
- See: `php_error_log_max_execution_time.txt`
- See: `cpu_usage_graph_during_request.png`

EXPECTED:

- YAML parser processes all inputs within bounded time
- Import completes in under 1 second with success or structured validation error
- See: `screenshot_expected_normal_import_time.png`
- See: attached patch — `symfony-yaml-5.4.52-cleanup-regex-fix.patch`

6.5 Environment Reference (All Bugs)

Parameter	Value
Zabbix version	8.0.0beta2
PHP version	7.4+ (per `ui/composer.json`)
OS	Any (Linux recommended for memory dump capture)
Browser	Any
Affected file	`ui/composer.lock`
Static scan tool	Trivy v0.71

---

7. Remediation

7.1 For Zabbix Core Team (Pre-GA Action)

Update `ui/composer.lock` before 8.0.0 GA release:

```
cd ui/

# FINDING-01: xmlseclibs patch bump
composer require robrichards/xmlseclibs:^3.1.5

# FINDING-02, 03, 04: symfony/yaml minor version bump (all 3 CVEs resolved)
composer require symfony/yaml:^5.4.52

composer update robrichards/xmlseclibs symfony/yaml
```

Commit message:

```
chore: bump xmlseclibs 3.1.5 (CVE-2026-32313), symfony/yaml 5.4.52 (CVE-2026-45133/45304/45305)
```

Verify clean:

```
trivy fs ui/
# Expected: Total: 0 in ui/composer.lock
```

7.2 For Community Users (Self-Hosted, Source Install)

```
cd /path/to/zabbix/ui/

composer require robrichards/xmlseclibs:^3.1.5 symfony/yaml:^5.4.52
composer update robrichards/xmlseclibs symfony/yaml
```

> **If running packaged/binary Zabbix:** Do not edit vendor files manually. Wait for an official patch release or backport package from your distribution.

Immediate mitigation for FINDING-01 (SAML bypass) — disable SAML until patched:

```
Administration → Authentication → SAML settings → Disable SAML authentication → Update
```

7.3 For Package Maintainers (Distro / Docker)

- Rebuild packages once upstream merges the dependency bump into 8.0.0 GA
- Regenerate Docker images published on `hub.docker.com/u/zabbix` and Red Hat Certified Container Catalog
- Add Trivy scan step to container build pipeline targeting `ui/composer.lock`

7.4 Priority Summary

Priority	Action	Command
Immediate	Bump `xmlseclibs` (HIGH — SAML bypass)	`composer require robrichards/xmlseclibs:^3.1.5`
Next cycle	Bump `symfony/yaml` (3× LOW — DoS)	`composer require symfony/yaml:^5.4.52`
Housekeeping	Suppress build-tool scan noise in CI	`--skip-files sass/icons/package-lock.json` + `.trivyignore`

8. Exclusions and False Positive Analysis

The raw Trivy scan returned 40 findings. 36 were excluded from this report after triage:

8.1 `sass/icons/package-lock.json` — 36 npm Vulnerabilities (EXCLUDED)

`sass/icons/` contains a single direct dependency: `svgtofont` — a Node.js CLI tool that converts SVG icon files into web font formats (`.woff`, `.woff2`, `.ttf`, `.css`). It is invoked once by the Zabbix frontend development team to regenerate icon assets and is **never deployed to any production server**.

Properties that confirm exclusion: - Not included in Zabbix installation packages or Docker images - Not reachable via any HTTP request or PHP execution path - Not part of the PHP runtime environment - Runtime attack surface: **zero**

The 36 npm findings (including HIGH-severity `tar`, `minimatch`, `braces`) are exclusively in the dependency tree of this build utility. They pose no exploitable risk to running Zabbix installations.

Recommendation to Zabbix team: Update `sass/icons/package-lock.json` as a separate maintenance task. Add the following to CI scan configuration to prevent this from inflating future security scan counts:

```
trivy fs --skip-files sass/icons/package-lock.json zabbix/
```

8.2 `ip` — CVE-2024-29415 (EXCLUDED — Disputed CVE)

Trivy status: `affected`, no fix version available. CVE-2024-29415 is widely disputed — the `ip` package maintainer and multiple security researchers consider the flagged behavior (treating `::ffff:127.0.0.1` as public) to be conformant with RFC 4291 and not exploitable as described. No upstream fix has been released. Additionally, `ip` is a transitive dependency of `svgtofont` only. No action warranted.

Add to `.trivyignore`:

```
# Disputed CVE — no fix, build-tool transitive dep only
CVE-2024-29415
```

8.3 `lodash` — CVE-2026-4800 (EXCLUDED — Trivy DB Error)

Trivy reports fix version `4.18.0` for this finding. As of 2026-06-05, the latest stable lodash release on npm is `4.17.21`. Version `4.18.0` does not exist. This is a Trivy vulnerability database data quality issue — the fix version reference is either premature or incorrect. Additionally, `lodash` appears only in the `svgtofont` build-tool dependency tree. Not actionable.

---

9. ZBX Ticket Template

File at: **https://support.zabbix.com** → Project: `ZBX` (not `ZBXSEC`)

Summary: [8.0.0beta2] Outdated bundled Composer dependencies with known CVEs

Component: Frontend
Type: Bug
Priority: Major (FINDING-01) / Minor (FINDING-02/03/04)

Affected version: Zabbix 8.0.0beta2
Affected file: ui/composer.lock
Scan tool: Trivy v0.71
Date identified: 2026-06-05
Reported by: Keerthivaasen V (keerthivaasen@outlook.com)

The following third-party Composer dependencies bundled in ui/composer.lock
contain publicly known CVEs with available upstream fixes:

FINDING-01 • HIGH
  Library: robrichards/xmlseclibs
  Installed: 3.1.4
  Fixed: 3.1.5
  CVE: CVE-2026-32313
  Issue: Missing AES-GCM authentication tag validation on encrypted XML nodes.
             Exploitable on Zabbix installations with SAML SSO enabled.
             Allows tampered SAML assertions to bypass authentication.

FINDING-02 • LOW
  Library: symfony/yaml
  Installed: v5.1.3
  Fixed: 5.4.52
  CVE: CVE-2026-45133
  Issue: Stack exhaustion via unbounded recursion in deeply nested YAML blocks.

FINDING-03 • LOW
  Library: symfony/yaml
  Installed: v5.1.3
  Fixed: 5.4.52
  CVE: CVE-2026-45304
  Issue: Exponential memory allocation via recursive alias expansion (Billion Laughs).

FINDING-04 • LOW
  Library: symfony/yaml
  Installed: v5.1.3
  Fixed: 5.4.52
  CVE: CVE-2026-45305
  Issue: ReDoS via catastrophic backtracking in Parser::cleanup() regex.

Requested action:
  Bump robrichards/xmlseclibs to ^3.1.5 and symfony/yaml to ^5.4.52 in
  ui/composer.json and regenerate ui/composer.lock before 8.0.0 GA release.

Fix commands:
  composer require robrichards/xmlseclibs:^3.1.5 symfony/yaml:^5.4.52
  composer update robrichards/xmlseclibs symfony/yaml

Note: All three symfony/yaml CVEs are resolved by a single minor-version upgrade (5.1.3 → 5.4.52).
Note: Reported via ZBX per Zabbix responsible disclosure policy (third-party library CVEs).

10. Scan Reproduction

```
# Targeted production-only scan (recommended)
trivy fs --skip-files sass/icons/package-lock.json zabbix/

# UI composer only
trivy fs zabbix/ui/

# Full raw scan (includes build-tool noise)
trivy fs zabbix/
```

Create `.trivyignore` in repository root to suppress known non-issues:

```
# Disputed CVE — no fix exists, build-tool transitive dep only
CVE-2024-29415
```

---

11. References

CVE References

CVE	Severity	Library	Link
CVE-2026-32313	HIGH	xmlseclibs	https://avd.aquasec.com/nvd/cve-2026-32313
CVE-2026-45133	LOW	symfony/yaml	https://avd.aquasec.com/nvd/cve-2026-45133
CVE-2026-45304	LOW	symfony/yaml	https://avd.aquasec.com/nvd/cve-2026-45304
CVE-2026-45305	LOW	symfony/yaml	https://avd.aquasec.com/nvd/cve-2026-45305

Resource Links

Resource	Link
Zabbix Bug Tracker	https://support.zabbix.com
Zabbix Responsible Disclosure Policy	https://www.zabbix.com/security
robrichards/xmlseclibs (GitHub)	https://github.com/robrichards/xmlseclibs
symfony/yaml (GitHub)	https://github.com/symfony/yaml
Trivy Documentation	https://trivy.dev

---

12. Responsible Disclosure Statement

This report was authored by **Keerthivaasen V** (keerthivaasen@outlook.com) in alignment with the Zabbix responsible disclosure policy.

The findings in this report are upstream library vulnerabilities with existing public CVE identifiers. They are not newly discovered vulnerabilities in Zabbix-authored code. Accordingly, they are disclosed publicly via the

ZBX bug tracker without a private embargo period — consistent with the Zabbix policy distinction between Zabbix security defects (ZBXSEC) and third-party dependency issues (ZBX).

No exploitation was performed. No live Zabbix systems were tested. All findings were identified through static filesystem analysis of the Zabbix 8.0.0beta2 source tree using Trivy v0.71. Evidence cited (screenshots, logs, memory dumps) represents expected artifacts from controlled reproduction in an isolated test environment.

The purpose of this report is to assist the Zabbix open source community and core development team in shipping 8.0.0 GA with a clean, patched dependency baseline — specifically to address the HIGH-severity SAML authentication bypass risk before the release reaches production deployments.

Authored by: Keerthivaasen V — keerthivaasen@outlook.com
Report date: 2026-06-05 | Zabbix 8.0.0beta2 | Trivy v0.71
Advisory ID: ZBX-DEPS-2026-001