Static Application Security Testing (SAST) Report

Organization: Caliber Technologies Inc. (Calimatic Mail) **Report Date:** February 14, 2026 **Tool:** npm audit (Node.js dependency scanner) **Scope:** All packages in the Calimatic Mail monorepo **Classification:** Confidential

---

1. Executive Summary

A static application security test was performed on the Calimatic Mail platform using `npm audit`, which scans all project dependencies against the GitHub Advisory Database for known vulnerabilities.

Metric	Value
Total Dependencies Scanned	1,193
Initial Vulnerabilities Found	37
Auto-Fixed (non-breaking)	20
Remaining After Remediation	17
Critical	0
High	6
Moderate	11
Low	0

Overall Risk Rating: Medium - All remaining vulnerabilities are in development/build-time dependencies or require major version upgrades that are scheduled for the next maintenance cycle.

2. Remediation Actions Taken

On February 14, 2026, `npm audit fix` was executed to automatically resolve all non-breaking vulnerabilities:

- **lodash** (Prototype Pollution via `_.unset` / `_.omit`) - Updated
- **qs** (arrayLimit bypass DoS) - Updated
- **fast-xml-parser** (RangeError DoS) - Updated
- **markdown-it** (ReDoS) - Updated
- **next** (DoS via Server Components, incomplete fix follow-up) - Partially updated

Result: 20 vulnerabilities resolved immediately.

3. Remaining Vulnerabilities

The following 17 vulnerabilities remain because their fixes require breaking (major version) changes. Each has been triaged and assessed for actual risk to the production environment.

3.1 HIGH Severity

H-1: Fastify DoS via sendWebStream (GHSA-mrq3-vjjr-p77c)

- **Package:** fastify <=5.7.2
- **Risk:** Unbounded memory allocation in sendWebStream
- **Impact Assessment:** LOW - The application does not use `sendWebStream`. Standard JSON responses are not affected.
- **Remediation Plan:** Upgrade to fastify@5.7.4 in next maintenance window.

H-2: Fastify Content-Type Tab Character Bypass (GHSA-jx2c-rxcm-jvmq)

- **Package:** fastify <=5.7.2

- **Risk:** Body validation bypass via tab character in Content-Type header
- **Impact Assessment:** LOW - All routes use Zod schema validation as a secondary defense layer. Rate limiting provides additional protection.
- **Remediation Plan:** Upgrade to fastify@5.7.4 in next maintenance window.

H-3: glob CLI Command Injection (GHSA-5j98-mcp5-4vw2)

- **Package:** glob 10.2.0 - 10.4.5 (via eslint-config-next)
- **Risk:** Command injection via -c/--cmd flag
- **Impact Assessment:** NONE - This is a development/lint dependency only. It is not included in production Docker images (multi-stage build). The CLI flag is never invoked by the application.
- **Remediation Plan:** Update eslint-config-next when Next.js 16 migration is completed.

H-4: Next.js Image Optimizer DoS (GHSA-9g9p-9gw9-jx7f)

- **Package:** next 10.0.0 - 15.5.9
- **Risk:** DoS via remotePatterns configuration
- **Impact Assessment:** LOW - Image optimization only allows localhost:9000 (MinIO). External image URLs are not accepted.
- **Remediation Plan:** Upgrade to Next.js 16 in scheduled migration.

H-5: Nodemailer Domain Interpretation Conflict (GHSA-mm7p-fcc7-pg87)

- **Package:** nodemailer <=7.0.10
- **Risk:** Email sent to unintended domain
- **Impact Assessment:** MEDIUM - Email sending uses validated, sanitized addresses. Outbound email goes through Haraka MTA with additional domain verification.
- **Remediation Plan:** Upgrade to nodemailer@8.x in next maintenance window.

H-6: Nodemailer addressparser DoS (GHSA-rcmh-qjqh-p98v)

- **Package:** nodemailer <=7.0.10
- **Risk:** DoS via recursive calls in address parsing
- **Impact Assessment:** LOW - Rate limiting (50 req/min write, 10 req/min sensitive) limits exploitation. Input validation sanitizes email addresses before they reach nodemailer.
- **Remediation Plan:** Upgrade to nodemailer@8.x in next maintenance window.

3.2 MODERATE Severity

M-1: esbuild Development Server Request Leak (GHSA-67mh-4wv8-2f99)

- **Package:** esbuild <=0.24.2 (via drizzle-kit, vite, vitest)
- **Impact Assessment:** NONE - esbuild is a build-time dependency only. The development server is never exposed in production. Not included in production Docker images.
- **Remediation Plan:** Will be resolved when drizzle-kit and vite are updated.

M-2: fast-jwt iss Claim Validation (GHSA-gm45-q3v2-6cf8)

- **Package:** fast-jwt <5.0.6 (via @fastify/jwt)
- **Impact Assessment:** LOW - The application uses HS256 with a server-controlled secret. The `iss` claim is not used for authorization decisions.
- **Remediation Plan:** Upgrade @fastify/jwt to v10 in next maintenance window.

4. Production Risk Assessment

Category	Count	Production Impact
Build/dev dependencies only (no production exposure)	11	NONE
Production dependencies with mitigating controls	6	LOW to MEDIUM

Key Mitigating Controls in Production:

- Multi-stage Docker builds exclude dev dependencies from production images
- Input validation (Zod schemas) on all API endpoints

- Rate limiting on all endpoint categories
- OAuth tokens never exposed in HTTP responses
- CSP, HSTS, and other security headers enforced
- TLS 1.3 for all traffic

5. Scheduled Remediation Timeline

Package	Current	Target	Scheduled
fastify	5.7.2	5.7.4+	Q1 2026
nodemailer	7.x	8.x	Q1 2026
@fastify/jwt	9.x	10.x	Q1 2026
next	15.x	16.x	Q2 2026
eslint-config-next	14.x	16.x	Q2 2026
drizzle-kit	current	latest	Q2 2026

6. Conclusion

The Calimatic Mail platform has **0 critical vulnerabilities** and all high-severity findings either affect build-time-only dependencies or have mitigating controls in production. Non-breaking fixes were applied immediately, reducing the total from 37 to 17. Remaining items are tracked and scheduled for remediation in upcoming maintenance windows.

Prepared by: Engineering Team, Caliber Technologies Inc. **Contact:** security@calimatic.app