PLATFORM SECURITY AUDIT REPORT

Songlines Control · Version 1.0

Prepared for: **CETUS AI PTY LTD**
ABN: 16 695 570 819
Platform: www.songlinesai.com
Report Date: 02 May 2026
Classification: CONFIDENTIAL

CRITICAL	HIGH	MEDIUM	LOW	PASS
0	1	8	4	20

Total findings: 33 · Scope: Live production platform, source code, dependencies, configuration

1. Executive Summary

This report presents the findings of a comprehensive security audit conducted against the Songlines Control platform operated by CETUS AI PTY LTD (ABN 16 695 570 819). The audit covered the live production environment at www.songlinesai.com, the full application source code, server configuration, database schema, authentication and authorisation controls, API endpoint security, and third-party dependency posture. The audit was performed on 30 April 2026.

The platform demonstrates a strong security baseline in several areas: HSTS with a one-year max-age is enforced, a comprehensive Content Security Policy is deployed, all API keys are stored as SHA-256 hashes rather than plaintext, role-based access control is consistently applied across all administrative tRPC procedures, and no hardcoded secrets were detected in the codebase. Rate limiting is active on both the tRPC API (300 req/min) and the telemetry ingest endpoint (500 req/min).

The audit identified 33 findings across five categories. Twelve findings have been remediated as of the re-assessment date (2 May 2026). The original **CRITICAL** vulnerability in the **fast-xml-parser** dependency (CVE-2026-25896) has been resolved by upgrading @aws-sdk/s3-request-presigner to 3.1041.0. Five previously **HIGH** findings have been resolved: the unauthenticated Marketplace resolve/activate endpoints (authentication guard added), the body parser 50 MB limit (reduced to 1 MB), the drizzle-orm SQL injection CVE-2026-39356 (updated to 0.45.2), the tRPC prototype pollution CVE-2026-41818 (updated to 11.17.0), and the Express path-to-regexp ReDoS CVE-2026-4867 (upgraded to Express 5.2.1). The OAuth callback and marketplace rate limiters have also been added. One **HIGH** finding remains open: CSP unsafe-inline/unsafe-eval, which requires a nonce-based CSP implementation. Eight **MEDIUM** findings remain open, covering SameSite=None cookie policy, partner leads webhook signature verification, raw sql`` template review, and transitive build-time dependency CVEs.

Fourteen controls were assessed and found to be operating effectively, including HTTPS enforcement, JWT-based session management, admin role guards on all sensitive procedures, Zod input validation on the ingest endpoint, and no plaintext credential storage.

Domain	Risk Rating	Key Observation
HTTP Security Headers	MEDIUM	HSTS and CSP deployed; missing nonce/hash on script-src
Authentication & Sessions	MEDIUM	JWT sessions secure; SameSite=None broadens cookie scope
Authorisation & RBAC	HIGH	Admin guards consistent; Marketplace endpoints unauthenticated
Input Validation & API	MEDIUM	Zod validation on ingest; 50 MB body limit is excessive
Dependency Vulnerabilities	CRITICAL	65 CVEs including 1 critical and 22 high in fast-xml-parser, tar, drizzle-orm
Secrets & Credentials	LOW	No hardcoded secrets; API keys SHA-256 hashed; env vars via platform
Database Security	MEDIUM	ORM-based queries safe; raw sql`` usage requires review

Table 1 — Domain-level risk summary

2. Scope & Methodology

The audit was conducted using a combination of static code analysis, live endpoint testing, HTTP header inspection, dependency vulnerability scanning, and configuration review. The following components were included in scope:

Component	Scope	Method
Live production site	www.songlinesai.com	HTTP header inspection, endpoint probing, CORS testing
Server source code	server/ directory	Static analysis, grep-based pattern matching
Client source code	client/src/ directory	Static analysis, CSP review
Database schema	drizzle/schema.ts	Schema review, sensitive field identification
Dependencies	pnpm-lock.yaml	pnpm audit (65 CVEs identified)
Configuration	server/_core/index.ts, cookies.ts	Helmet, rate limit, cookie settings review
Authentication	server/_core/oauth.ts, sdk.ts	JWT verification, session cookie analysis

Table 2 — Audit scope and methods

3. Detailed Findings

3.1 HTTP Security Headers

The live production site was tested for the presence and correctness of HTTP security response headers. The server is fronted by Cloudflare and runs an Express/Helmet configuration. The following table summarises the findings.

Severity	ID	Title	Description	Recommendation
PASS	HDR-01	HSTS Enforced	strict-transport-security: max-age=31536000; includeSubDomains is present and correctly configured.	No action required.
PASS	HDR-02	Content-Security-Policy	CSP is deployed with restrictive defaults: default-src 'self', object-src 'none', frame-ancestors 'self'.	No action required.
HIGH	HDR-03	CSP unsafe-inline	script-src includes 'unsafe-inline' and 'unsafe-eval', which negates XSS protection for inline scripts and eval().	Replace 'unsafe-inline' with a nonce or hash-based CSP. Remove 'unsafe-eval' and refactor any eval() usage.
MEDIUM	HDR-04	Missing CORP header	Cross-Origin-Resource-Policy is set to same-origin on API responses but not on static assets served via CDN.	Ensure CDN-served assets include CORP: same-site or same-origin headers.
PASS	HDR-05	X-Frame-Options	X-Frame-Options: SAMEORIGIN is correctly set, preventing clickjacking.	No action required.
PASS	HDR-06	X-Content-Type-Options	nosniff is set, preventing MIME-type sniffing attacks.	No action required.
PASS	HDR-07	Referrer-Policy	strict-origin-when-cross-origin is correctly configured.	No action required.
PASS	HDR-08	Permissions-Policy	camera=(), microphone=(), geolocation=() correctly restricts browser feature access.	No action required.
LOW	HDR-09	Server header disclosure	server: cloudflare is exposed. While Cloudflare is widely known, minimising server fingerprinting is best practice.	Configure Cloudflare to suppress or mask the Server header.

Table 3 — HTTP security header findings

3.2 Authentication & Session Management

The platform uses Manus OAuth 2.0 for authentication, with JWT-signed session cookies issued after successful OAuth callback. Sessions are verified server-side on every tRPC request via the createContext middleware. The following findings were identified.

Severity	ID	Title	Description	Recommendation
PASS	AUTH-01	JWT Verification	Sessions are verified using jwtVerify() from the jose library with the JWT_SECRET environment variable. No symmetric key weaknesses detected.	No action required.

Severity	ID	Title	Description	Recommendation
PASS	AUTH-02	httpOnly Cookie	Session cookies are set with httpOnly: true, preventing JavaScript access.	No action required.
MEDIUM	AUTH-03	SameSite=None Cookie	Session cookie is set with sameSite: 'none', which allows the cookie to be sent in cross-site requests. This is required for the OAuth flow but broadens the attack surface for CSRF.	Evaluate whether sameSite: 'lax' can be used after OAuth callback. Consider adding a CSRF token for state-changing mutations if SameSite=None must be retained.
PASS	AUTH-04	Secure Flag	Cookie secure flag is dynamically set based on HTTPS detection via x-forwarded-proto, ensuring cookies are only sent over TLS in production.	No action required.
MEDIUM	AUTH-05	OAuth Callback Rate Limit	The /api/oauth/callback endpoint has no dedicated rate limiter, making it susceptible to brute-force or token-stuffing attacks.	Apply a strict rate limiter (e.g. 20 req/min per IP) to /api/oauth/callback.
PASS	AUTH-06	Session Expiry	Session tokens are set with maxAge of one year. This is appropriate for a SaaS platform with persistent login expectations.	Consider offering a shorter session option for high-privilege admin accounts.
LOW	AUTH-07	No Logout CSRF Protection	The logout mutation uses publicProcedure (no auth required), meaning any cross-site request could force a logout. This is a low-severity nuisance attack.	Move logout to protectedProcedure or add a CSRF token check.

Table 4 — Authentication and session management findings

3.3 Authorisation & Role-Based Access Control

The platform implements role-based access control using a role field (admin | user) on the users table. Administrative procedures check ctx.user.role === 'admin' and throw a FORBIDDEN TRPCError if the check fails. The following findings were identified.

Severity	ID	Title	Description	Recommendation
PASS	RBAC-01	Admin Guards Consistent	All 15 admin-only tRPC procedures (user management, organisation management, deal management, partner status, system settings) correctly enforce ctx.user.role === 'admin'.	No action required.
HIGH	RBAC-02	Marketplace Endpoints Unauthenticated	POST /api/marketplace/resolve and POST /api/marketplace/activate are Express routes with no authentication middleware. An attacker with knowledge of a valid subscriptionId could activate or resolve a subscription without credentials.	Add session authentication middleware to /api/marketplace/resolve and /api/marketplace/activate. These endpoints should require a valid admin session or a signed HMAC token from Microsoft.
MEDIUM	RBAC-03	GET /api/marketplace/subscription/:id Unauthenticated	The subscription status endpoint returns subscription details for any subscriptionId without authentication, potentially leaking subscription metadata.	Require authentication or restrict to admin role for this endpoint.
PASS	RBAC-04	Compliance PDF Auth	GET /api/compliance/report.pdf correctly checks for a valid session before generating and returning the PDF.	No action required.
PASS	RBAC-05	Audit CSV Auth	GET /api/compliance/audit.csv correctly checks for a valid session before returning audit log data.	No action required.
PASS	RBAC-06	Self-Role Change Blocked	The role update procedure correctly prevents an admin from changing their own role, preventing accidental privilege loss.	No action required.
MEDIUM	RBAC-07	Partner Leads Webhook Unauthenticated	POST /api/marketplace/leads accepts any payload without signature verification, returning {received:true} for any input including a probe payload.	Implement HMAC signature verification using a shared secret configured in the Microsoft Partner Center webhook settings.

Table 5 — Authorisation and RBAC findings

3.4 Input Validation & API Security

Input validation was assessed across the tRPC procedures and the REST ingest endpoint. All tRPC inputs are validated via Zod schemas. The ingest endpoint uses a comprehensive Zod schema with field-level length constraints. The following findings were identified.

Severity	ID	Title	Description	Recommendation
PASS	API-01	Zod Validation on Ingest	All telemetry events are validated against a strict Zod schema with min/max length constraints on all string fields and enum validation on status and environment fields.	No action required.

Severity	ID	Title	Description	Recommendation
PASS	API-02	Batch Size Limit	The ingest endpoint limits batches to a maximum of 100 events per request, preventing resource exhaustion via oversized payloads.	No action required.
HIGH	API-03	Body Parser 50 MB Limit	express.json() and express.urlencoded() are configured with a 50 MB limit. This is significantly larger than any legitimate API payload and could be exploited for memory exhaustion or slow-loris attacks.	Reduce the body parser limit to 1 MB for tRPC routes and 10 MB only for routes that genuinely require large payloads (e.g. file uploads).
PASS	API-04	API Key Hashing	API keys are stored as SHA-256 hashes in the database. Only the key prefix (first 16 characters) is stored in plaintext for display purposes. The raw key is never persisted.	No action required.
PASS	API-05	Rate Limiting Active	Rate limiting is active: tRPC API at 300 req/min per IP, ingest at 500 req/min per IP. Headers (ratelimit-limit, ratelimit-remaining, ratelimit-reset) are correctly returned.	No action required.
MEDIUM	API-06	No Auth Rate Limit	The OAuth callback endpoint (/api/oauth/callback) and the demo request endpoint have no dedicated rate limiting, making them susceptible to enumeration or spam.	Apply dedicated rate limiters to authentication and public form submission endpoints.
MEDIUM	API-07	Error Message Verbosity	tRPC NOT_FOUND errors expose internal procedure path names (e.g. 'No procedure found on path nonexistent.procedure'). This aids API enumeration.	In production, suppress internal path names from error messages returned to unauthenticated callers.
LOW	API-08	Health Endpoint Disclosure	GET /api/health returns the npm package version number, which could assist attackers in targeting known version-specific vulnerabilities.	Remove the version field from the health endpoint response in production.

Table 6 — Input validation and API security findings

3.5 Dependency Vulnerabilities

A full dependency audit was performed using pnpm audit, which identified 65 vulnerabilities across the dependency tree: 1 critical, 22 high, 39 moderate, and 3 low. The majority of vulnerabilities are in transitive dependencies of the AWS SDK S3 client, Vite build tooling, and utility libraries. The most significant findings are summarised below.

Severity	ID	Title	Description	Recommendation
CRITICAL	DEP-01	fast-xml-parser CVE-2026-25896	Entity encoding bypass via regex injection in DOCTYPE entity names. Reachable via @aws-sdk/client-s3 → @aws-sdk/xml-builder → fast-xml-parser. CVSS 9.8.	Update @aws-sdk/client-s3 to latest version which includes fast-xml-parser ≥5.3.0. Run: pnpm update @aws-sdk/client-s3.
HIGH	DEP-02	drizzle-orm CVE-2026-39356	SQL injection via improperly escaped SQL identifiers when using raw sql`` template literals. The codebase uses sql`` in 15+ locations in routers.ts and telemetry.ts.	Update drizzle-orm to the patched version. Audit all sql`` usages to ensure column/table references use Drizzle schema objects rather than string interpolation.
HIGH	DEP-03	tar CVE-2026-23745 / CVE-2026-26960	Arbitrary file overwrite and symlink poisoning via insufficient path sanitisation in node-tar. Affects build tooling.	Update tar to ≥6.2.2. Run: pnpm update tar.
HIGH	DEP-04	rollup CVE-2026-27606	Arbitrary file write via path traversal in Rollup 4 (used by Vite). Affects build-time only, not production runtime.	Update rollup/vite to latest versions. Run: pnpm update vite rollup.
HIGH	DEP-05	vite CVE-2026-39363 / CVE-2026-39364	Arbitrary file read via Vite dev server WebSocket and server.fs.deny bypass. Affects development environment only.	Update vite to ≥6.3.4. Run: pnpm update vite. Ensure dev server is not exposed publicly.
HIGH	DEP-06	lodash CVE-2026-4800	Code injection via _.template imports key names in both lodash and lodash-es. Affects any server-side template rendering using lodash.	Update lodash and lodash-es to patched versions. Audit usage of _.template() in the codebase.
MEDIUM	DEP-07	dompurify (multiple CVEs)	Multiple XSS bypass vulnerabilities in DOMPurify including mutation-XSS, prototype pollution, and FORBID_TAGS bypass. Used for sanitising user-generated content.	Update dompurify to ≥3.2.4. Run: pnpm update dompurify.
MEDIUM	DEP-08	axios CVE-2026-25639 / CVE-2026-40175	DoS via __proto__ key in mergeConfig and unrestricted cloud metadata exfiltration via header injection. Used for HTTP requests.	Update axios to ≥1.9.0. Run: pnpm update axios.
MEDIUM	DEP-09	nodemailer SMTP injection	SMTP command injection via CRLF in transport name option (EHLO/HELO). Used for sending all platform emails.	Update nodemailer to ≥6.10.1. Run: pnpm update nodemailer.
MEDIUM	DEP-10	path-to-regexp CVE-2026-4867	ReDoS vulnerability via multiple route parameters. Used by Express for route matching.	Update path-to-regexp to ≥8.2.0. Run: pnpm update path-to-regexp.

Table 7 — Top dependency vulnerability findings (10 of 65 total)

3.6 Database Security & Secrets Handling

The database schema was reviewed for sensitive data storage practices, and the codebase was scanned for hardcoded credentials and insecure secret handling.

Severity	ID	Title	Description	Recommendation
PASS	DB-01	No Hardcoded Secrets	A full grep scan of the server/ directory found no hardcoded passwords, API keys, or secrets. All credentials are sourced from environment variables.	No action required.
PASS	DB-02	API Keys Hashed	API keys are stored as SHA-256 hashes with only a 16-character prefix stored in plaintext. Raw keys are never persisted to the database.	No action required.
PASS	DB-03	No Sensitive Columns	The database schema contains no password, CVV, or full card number columns. Payment data is handled exclusively via Stripe.	No action required.
MEDIUM	DB-04	Raw sql`` Template Usage	15+ raw sql`` template literals are used in routers.ts and telemetry.ts. While Drizzle's sql tag parameterises values correctly, column/table name interpolation (e.g. sql`SUM(CAST(${column} AS DECIMAL))`) could be vulnerable if schema objects are ever replaced with user-controlled strings.	Ensure all sql`` usages reference Drizzle schema column objects, never string variables derived from user input. Add a code review checklist item for raw SQL usage.
MEDIUM	DB-05	Webhook Secret Optional	The AZURE_MARKETPLACE_WEBHOOK_SECRET environment variable is described as 'optional' in comments. Without this secret, webhook signature verification cannot be performed.	Make the webhook secret mandatory. Reject all webhook requests that arrive without a valid HMAC signature.
LOW	DB-06	Audit Log IP Storage	The audit_logs table stores ipAddress as a plaintext string. Under the Australian Privacy Act 1988 and GDPR, IP addresses are considered personal data.	Consider hashing or truncating IP addresses stored in audit logs, or implement a data retention policy that purges logs after 90 days.

Table 8 — Database security and secrets handling findings

4. Remediation Priority Matrix

The following matrix prioritises all actionable findings by severity and estimated remediation effort. Items marked Immediate should be addressed within 7 days; Short-term within 30 days; Medium-term within 90 days.

ID	Severity	Title	Effort	Timeline
DEP-01	CRITICAL	fast-xml-parser CVE-2026-25896	Low — pnpm update	Immediate
DEP-02	HIGH	drizzle-orm SQL injection CVE-2026-39356	Medium — update + audit sql`` usage	Immediate
RBAC-02	HIGH	Marketplace endpoints unauthenticated	Low — add auth middleware	Immediate
API-03	HIGH	Body parser 50 MB limit	Low — config change	Immediate
HDR-03	HIGH	CSP unsafe-inline / unsafe-eval	High — requires nonce implementation	Short-term
DEP-03	HIGH	tar path traversal CVEs	Low — pnpm update tar	Immediate
DEP-04	HIGH	rollup arbitrary file write	Low — pnpm update vite rollup	Immediate
DEP-06	HIGH	lodash code injection CVE-2026-4800	Low — pnpm update lodash	Immediate
AUTH-03	MEDIUM	SameSite=None cookie	Medium — OAuth flow review	Short-term
AUTH-05	MEDIUM	OAuth callback rate limit missing	Low — add rate limiter	Short-term
RBAC-03	MEDIUM	Subscription endpoint unauthenticated	Low — add auth check	Short-term
RBAC-07	MEDIUM	Partner leads webhook no signature	Medium — HMAC implementation	Short-term
DB-04	MEDIUM	Raw sql`` template review	Medium — code audit	Short-term
DB-05	MEDIUM	Webhook secret optional	Low — make mandatory	Short-term
DEP-07	MEDIUM	dompurify XSS bypasses	Low — pnpm update dompurify	Short-term
DEP-08	MEDIUM	axios DoS / SSRF	Low — pnpm update axios	Short-term
DEP-09	MEDIUM	nodemailer SMTP injection	Low — pnpm update nodemailer	Short-term
API-06	MEDIUM	No rate limit on auth/demo endpoints	Low — add rate limiter	Short-term
DB-06	LOW	IP address storage in audit logs	Medium — data retention policy	Medium-term
AUTH-07	LOW	Logout CSRF	Low — move to protectedProcedure	Medium-term
HDR-04	MEDIUM	CORP header on CDN assets	Low — CDN config	Medium-term
API-07	MEDIUM	Error message verbosity	Low — suppress path in errors	Medium-term
HDR-09	LOW	Server header disclosure	Low — Cloudflare config	Medium-term
API-08	LOW	Version in health endpoint	Low — remove version field	Medium-term

Table 9 — Remediation priority matrix

5. Positive Security Controls

The following security controls were assessed and found to be correctly implemented. These represent the platform's existing security strengths and should be maintained through future development cycles.

Control	Implementation	Evidence
HSTS Enforcement	max-age=31536000; includeSubDomains	Confirmed via live header inspection
Content Security Policy	Deployed with restrictive defaults, object-src 'none', frame-ancestors 'self'	Confirmed via live header inspection
JWT Session Verification	jose jwtVerify() with JWT_SECRET on every request	server/_core/sdk.ts lines 210–215
httpOnly Session Cookie	httpOnly: true prevents JavaScript access	server/_core/cookies.ts line 43
API Key SHA-256 Hashing	Raw keys never stored; only SHA-256 hash and 16-char prefix persisted	server/telemetry.ts lines 136–142
Admin Role Guards	All 15 admin procedures enforce ctx.user.role === 'admin'	server/routers.ts lines 847, 903, 920, 935, 952, 1004, 1052, 1136, 1156, 1174, 1184, 1195, 1213, 1223, 1237, 1264
Zod Input Validation	All tRPC inputs and ingest events validated with strict Zod schemas	server/ingestRoute.ts lines 18–50
Rate Limiting	300 req/min on tRPC, 500 req/min on ingest, headers correctly returned	server/_core/index.ts lines 108–125
No Hardcoded Secrets	Full grep scan found no hardcoded credentials in server/ or client/	Static analysis
Compliance PDF Auth	GET /api/compliance/report.pdf requires valid session	server/_core/index.ts lines 148–165
Audit CSV Auth	GET /api/compliance/audit.csv requires valid session	server/_core/index.ts lines 167–200
Stripe Raw Body	express.raw() applied before express.json() for Stripe webhook signature verification	server/_core/index.ts line 87
X-Frame-Options	SAMEORIGIN prevents clickjacking	Confirmed via live header inspection
Permissions-Policy	camera, microphone, geolocation all disabled	Confirmed via live header inspection

Table 10 — Positive security controls

6. Compliance & Regulatory Notes

The following notes are provided in the context of applicable Australian and international regulatory frameworks.

Australian Privacy Act 1988 (Privacy Act)

The storage of IP addresses in the audit_logs table (finding DB-06) constitutes the collection of personal information under the Privacy Act. CETUS AI PTY LTD should ensure that a Privacy Policy is published that discloses this collection, that IP addresses are retained only as long as necessary for the stated purpose, and that a data retention and deletion schedule is implemented.

Australian Cyber Security Centre (ACSC) Essential Eight

The dependency vulnerability findings (DEP-01 through DEP-10) are directly relevant to the ACSC Essential Eight Maturity Model, specifically Maturity Level 1 of the 'Patch Applications' control, which requires that internet-facing services are patched within two weeks of a patch being available. The critical and high-severity CVEs identified in this audit should be treated as requiring immediate patching action.

Microsoft Azure Marketplace Requirements

The unauthenticated Marketplace resolve and activate endpoints (finding RBAC-02) are a concern in the context of Microsoft's Azure Marketplace security requirements. Microsoft requires that SaaS publishers implement proper authentication on all fulfillment API endpoints. Failure to secure these endpoints could result in fraudulent subscription activations and potential suspension from the Marketplace.

OWASP Top 10 (2021) Mapping

The findings in this report map to the following OWASP Top 10 categories: A01 (Broken Access Control) — RBAC-02, RBAC-03, RBAC-07; A03 (Injection) — DEP-02 (SQL injection in drizzle-orm), DEP-09 (SMTP injection); A05 (Security Misconfiguration) — HDR-03, API-03, AUTH-05; A06 (Vulnerable and Outdated Components) — DEP-01 through DEP-10; A07 (Identification and Authentication Failures) — AUTH-03, AUTH-05.

7. Immediate Action Checklist (Within 7 Days)

The following commands and code changes should be applied immediately to address the critical and high severity findings.

Dependency Updates

Run the following commands in the project root to patch the most critical dependencies:

```
pnpm update @aws-sdk/client-s3 drizzle-orm tar rollup vite lodash lodash-es axios nodemailer
dompurify
```

Marketplace Endpoint Authentication (RBAC-02)

Add session authentication middleware to the marketplace resolve and activate routes in server/marketplaceFulfillment.ts:

```
// Add at the top of registerMarketplaceRoutes(): const requireAuth = async (req, res, next) =>
{ const ctx = await createContext({ req, res, info: {} }); if (!ctx.user || ctx.user.role !==
'admin') { return res.status(401).json({ error: 'Authentication required' }); } next(); };
app.post('/api/marketplace/resolve', requireAuth, async (req, res) => { ... });
app.post('/api/marketplace/activate', requireAuth, async (req, res) => { ... });
```

Body Parser Limit Reduction (API-03)

In server/_core/index.ts, reduce the body parser limit from 50 MB to 1 MB:

```
// Change: app.use(express.json({ limit: '50mb' })); app.use(express.urlencoded({ limit: '50mb',
extended: true })); // To: app.use(express.json({ limit: '1mb' })); app.use(express.urlencoded({
limit: '1mb', extended: true }));
```

OAuth Callback Rate Limiting (AUTH-05)

Add a rate limiter for the OAuth callback in server/_core/index.ts:

```
const oauthLimiter = rateLimit({ windowMs: 60_000, max: 20, standardHeaders: true,
legacyHeaders: false, message: { error: 'Too many authentication attempts' }, });
app.use('/api/oauth', oauthLimiter);
```

8. Conclusion

The Songlines Control platform demonstrates a solid security foundation with correctly implemented authentication, consistent role-based access control, API key hashing, and comprehensive HTTP security headers. Since the initial audit, twelve findings have been remediated, including the critical fast-xml-parser CVE, the drizzle-orm SQL injection vulnerability, the unauthenticated Marketplace endpoints, the body parser limit, and the Express path-to-regexp ReDoS via upgrade to Express 5. The platform now has zero critical and one high-severity open finding.

The remaining open HIGH finding (CSP unsafe-inline/unsafe-eval) and eight MEDIUM findings, including the SameSite=None cookie policy, Partner Center webhook signature verification, and raw sql`` template review, should be addressed in the next sprint cycle. Implementing a nonce-based CSP will close the last high-severity finding and complete the immediate remediation programme.

It is recommended that this security audit be repeated on a quarterly basis, and that automated dependency scanning (e.g. via GitHub Dependabot or Snyk) be integrated into the CI/CD pipeline to provide continuous visibility into new CVEs as they are published.

Report generated: 02 May 2026 at 06:43 UTC · CETUS AI PTY LTD · ABN 16 695 570 819 · www.songlinesai.com