

Security Research Report: Source Review

Redacted public version · Report date 2026-10-25 · Version 3.0

Redaction notice. Route paths, parameter and file names, identifiers, code excerpts, and exact dependency versions have been removed. Findings are described at the pattern level so the report shows method and judgment without publishing a map of a specific deployment.

1. Scope and posture

Target: source archive of a multi-tenant AI messaging application (the author own project). **Perspective:** external or low-privilege attacker. **Method:** static source review only. No live target, network request, real account, token, or exploit was used, and findings that depend on deployment or runtime behavior are marked unconfirmed.

Overall assessment: High-priority candidate findings, plus one conditional critical dependency item that needs applicability validation. These are source-derived leads, not confirmed exploits.

2. Findings summary

ID	Finding	Severity
001	Unauthenticated read of conversation messages by identifier	High (provisional)
002	Conversation not bound to agent or tenant on public write	High (provisional)
003	Webhook verification secret written to application logs	Medium
004	Process-local rate limiting and trust of forwarded client IP headers	Low–Medium
005	Content Security Policy allows unsafe script directives	Low
006	Type-check failures ignored at build time	Informational
007	Cross-tenant agent reassignment through connection update	High
008	Server-side request forgery in URL-based knowledge ingestion	High (provisional)
009	Unauthenticated email-notification webhook can trigger mailbox polling	Medium–High
010	OAuth callbacks trust unsigned, modifiable state	Medium–High
011	Dependency versions inside a published critical advisory range	Critical (conditional)
012	Cron endpoint authorization can fail open when its secret is unset	Medium (conditional)
013	Connection update response returns stored OAuth tokens to the browser	Low–Medium
014	OAuth tokens stored in plaintext; encryption helper unused	Medium

Breakdown: 1 conditional critical, 4 high, 2 medium–high, 3 medium (one conditional), 2 low–medium, 1 low, 1 informational. "Provisional" means severity depends on deployment validation.

3. Methodology

1. Identify remotely exposed entry points and public routes.
2. Trace attacker-controlled identifiers, URLs, state values, and bodies into database operations and outbound requests.
3. Check that ownership, tenant, session, webhook-authenticity, and OAuth-state checks run before sensitive actions.
4. Review secrets and token material in responses, storage, and logs.
5. Compare locked dependency versions with upstream advisories and assess whether the affected feature is enabled.
6. Separate source observations from behavior that needs live validation.

4. Candidate findings

001 — Unauthenticated read of conversation messages by identifier

Severity: High (provisional) · **CWE:** CWE-862, CWE-639 · **OWASP WSTG:** WSTG-ATHZ-04 · **Status:** source-derived, runtime unconfirmed

Observation. A public endpoint returns stored messages for a caller-supplied conversation identifier. No session, ownership, or tenant check is visible in the handler, and the route is marked public in middleware.

Impact. Possible disclosure of customer conversation content to anyone holding a valid identifier. Real exposure depends on identifier entropy and data sensitivity.

Remediation. Require a session for dashboard reads and a scoped, short-lived token for public widgets. Verify ownership before querying, return a non-disclosing 404, and add regression tests for unauthenticated, cross-user, and cross-tenant requests.

Validation. Unauthenticated and cross-tenant reads fail; the owner can still read their own conversation.

002 — Conversation not bound to agent or tenant on public write

Severity: High (provisional) · **CWE:** CWE-862, CWE-639, CWE-20 · **OWASP WSTG:** WSTG-ATHZ-02, WSTG-ATHZ-04 · **Status:** source-derived, runtime unconfirmed

Observation. The public message-submission path accepts an agent identifier and a conversation identifier independently. It checks that each exists but not that they belong together.

Impact. Cross-tenant message injection, or processing one tenant history under another tenant agent instructions.

Remediation. Verify the conversation-to-agent and conversation-to-tenant relationship before any read, write, or model execution. Use a scoped widget credential, set input length limits, and test that mismatched identifiers cause no side effects.

Validation. Mismatched combinations are rejected before any write; valid widget traffic still works.

003 — Webhook verification secret written to application logs

Severity: Medium · **CWE:** CWE-532 · **OWASP WSTG:** WSTG-ERRH-01, WSTG-CONF-02 · **Status:** source-derived, runtime unconfirmed

Observation. The webhook verification handler logs the supplied token, the configured secret, and the challenge value.

Impact. Anyone with log access can learn the secret. Practical reuse and log access were not tested.

Remediation. Remove the logging, record only the outcome and a correlation ID, restrict log access, and rotate the secret if retained logs may have exposed it.

Validation. With synthetic values in a local environment, captured logs contain no secret or challenge.

004 — Process-local rate limiting and trust of forwarded client IP headers

Severity: Low–Medium · **CWE:** CWE-770 · **OWASP WSTG:** — · **Status:** source-derived, runtime unconfirmed

Observation. Rate-limit counters live in process memory, and the client IP is derived from forwarding headers.

Impact. Counters can be inconsistent across instances, and a client may influence its own limit key if the proxy does not overwrite untrusted headers. Neither condition was verified.

Remediation. Use a shared store or gateway limiter, trust forwarding headers only from known proxies, and add route-specific limits for expensive operations.

Validation. Identity and counters are stable across instances in a small staging test.

005 — Content Security Policy allows unsafe script directives

Severity: Low · **CWE:** CWE-693 · **OWASP WSTG:** — · **Status:** source-derived, runtime unconfirmed

Observation. The script policy permits inline scripts and dynamic code evaluation.

Impact. Weakens defense in depth. No XSS was confirmed.

Remediation. Move to nonces or hashes, remove dynamic evaluation in production if not required, and trial a stricter report-only policy first.

Validation. Production headers are inspected, and a controlled injection fixture is blocked in staging.

006 — Type-check failures ignored at build time

Severity: Informational · **CWE:** — · **OWASP WSTG:** — · **Status:** source-derived, runtime unconfirmed

Observation. The build configuration ignores type errors.

Impact. Reduces build-time assurance. Not itself a runtime flaw.

Remediation. Fail CI on type errors and add authorization regression tests.

Validation. CI fails on a deliberate type error in a disposable branch.

007 — Cross-tenant agent reassignment through connection update

Severity: High · **CWE:** CWE-639, CWE-862 · **OWASP WSTG:** — · **Status:** source-derived, runtime unconfirmed

Observation. The update handlers for messaging connections and integrations correctly verify that the connection belongs to the caller. They then accept a new agent identifier without verifying that the agent belongs to the same caller. Later message processing uses the stored agent.

Impact. An authenticated tenant could route its own inbound messages through another tenant agent, potentially invoking it and disclosing its instructions or knowledge through replies. End-to-end impact was not dynamically tested.

Remediation. Verify agent ownership on every create, update, OAuth callback, and webhook-derived association; reject foreign identifiers; add a regression test for one tenant attaching a connection to another tenant agent.

Validation. The update is rejected with no side effects when the agent belongs to a different tenant.

008 — Server-side request forgery in URL-based knowledge ingestion

Severity: High (provisional) · **CWE:** CWE-918 · **OWASP WSTG:** — · **Status:** source-derived, runtime unconfirmed

Observation. Authenticated users can submit a URL that the server fetches after only a syntax check. There is no visible destination allowlist, private-address blocking, redirect control, or response-size cap.

Impact. The server could be induced to request internal services or cloud metadata endpoints. Fetched content is stored and readable through the tenant own agent, and very large responses can exhaust memory. Actual impact depends on hosting network isolation.

Remediation. Allow public HTTP(S) destinations only. Validate DNS answers and resolved IPs against loopback, private, link-local, multicast, and metadata ranges; bind the connection to the validated address; disable or revalidate redirects; enforce size, content-type, and timeout limits; isolate fetching in a constrained egress service where arbitrary URLs are required.

Validation. In a controlled network, internal targets and redirect chains are rejected while an allowed public URL still works.

009 — Unauthenticated email-notification webhook can trigger mailbox polling

Severity: Medium–High · **CWE:** CWE-306, CWE-799 · **OWASP WSTG:** — · **Status:** source-derived, runtime unconfirmed

Observation. The handler accepts a caller-supplied payload, looks up a connected account by address, and triggers polling. It does not verify a provider-signed token or shared secret, unlike the other webhook in the same application.

Impact. Unauthorized triggering of processing that consumes provider quota, model usage, and compute. No inbox contents are returned directly.

Remediation. Enable authenticated push delivery and validate the token signature, issuer, expiry, expected service account, and audience. Add a shared secret as defense in depth, rate-limit deliveries, and reject malformed or oversized payloads.

Validation. Requests without valid provider authentication are rejected before any database lookup or polling.

010 — OAuth callbacks trust unsigned, modifiable state

Severity: Medium–High · **CWE:** CWE-352, CWE-345 · **OWASP WSTG:** — · **Status:** source-derived, runtime unconfirmed

Observation. The OAuth state value is plain encoded JSON carrying user and agent identifiers. It is not integrity-protected, not compared to the active session, not backed by a server-side nonce, and agent ownership is not verified.

Impact. A crafted callback could link an external account to a different user or agent, if an attacker can obtain a valid provider code and the callback runs in a logged-in browser. Amplified by finding 007; the combined flow was not tested.

Remediation. Generate a random, single-use state tied to the initiating session and stored server-side. Expire it, compare the callback session to the stored user, verify agent ownership before persisting tokens, and use PKCE where supported.

Validation. Tampered, replayed, cross-session, and foreign-agent states all fail before token persistence.

011 — Dependency versions inside a published critical advisory range

Severity: Critical (conditional) · **CWE:** — · **OWASP WSTG:** — · **Status:** source-derived, runtime unconfirmed

Observation. The locked versions of the web framework and an image-processing library fall within the affected range of a published critical advisory for the image-optimization surface. The application configuration disables the optimizer, which appears to remove that path. Exact versions and advisory identifiers are omitted here.

Impact. Potential remote code execution if the optimizer is reachable in deployment. Not confirmed as exploitable.

Remediation. Upgrade both packages to patched releases and rebuild from a clean lockfile. Do not rely on the configuration flag as the long-term fix. Confirm production platform and reachability.

Validation. A dependency audit against the deployed lockfile shows patched versions, and the optimizer route behaves as expected in staging.

012 — Cron endpoint authorization can fail open when its secret is unset

Severity: Medium (conditional) · **CWE:** CWE-306 · **OWASP WSTG:** — · **Status:** source-derived, runtime unconfirmed

Observation. The handler compares the request header to a string built from an environment variable. If the variable is absent, the expected value becomes a predictable string.

Impact. With the secret missing, a remote caller could trigger polling for every connected account, with workflow and provider-cost impact.

Remediation. Fail closed when the secret is absent or too short, preferably at startup, and keep the endpoint behind platform cron authentication.

Validation. With the variable removed in a controlled environment, the endpoint returns a configuration error and does not run the job.

013 — Connection update response returns stored OAuth tokens to the browser

Severity: Low–Medium · **CWE:** CWE-200 · **OWASP WSTG:** — · **Status:** source-derived, runtime unconfirmed

Observation. The update handler returns the full stored record, including access and refresh tokens. The request is owner-authenticated, so this is not cross-user theft on its own.

Impact. Unnecessarily moves server-held credentials into client-accessible data.

Remediation. Return a narrow safe projection and never include token fields. Rotate credentials if they were exposed beyond the intended owner.

Validation. Response JSON contains no token or secret fields.

014 — OAuth tokens stored in plaintext; encryption helper unused

Severity: Medium · **CWE:** CWE-312 · **OWASP WSTG:** — · **Status:** source-derived, runtime unconfirmed

Observation. The data model notes that integration tokens are stored as-is, and OAuth callbacks and integration creation persist them directly. An AES-256-GCM helper exists but has no call sites.

Impact. A database dump, backup, or read-only database compromise could expose credentials for linked accounts, depending on scope and expiry.

Remediation. Encrypt tokens before persistence with a managed key stored separately from the database, migrate existing values, decrypt only inside server-side services, and redact secrets from logs and responses.

Validation. Newly stored values are ciphertext, integrations keep working, and no response or log exposes tokens.

5. Controls observed

- Session-based authentication is used by some routes, and some resource handlers scope queries to the current user.
- Input validation is present on some endpoints.
- One webhook verifies request signatures.
- Security headers are configured, and deprecated custom authentication routes are disabled.

These observations do not prove that all routes or production settings are secure.

6. Limitations

Not performed: live HTTP testing, runtime logs, fuzzing, load testing, credential validation, or any third-party integration. A registry-based dependency audit was not completed. SQL injection, XSS, file upload, CORS, cookie flags, and IDOR across other routes were not confirmed. This is a source-assisted lead report, not a validated bounty submission, and no finding should be called live-confirmed until deployed behavior and impact are demonstrated on an in-scope target.