SECURITY POSTURE · V2026.04

ADAWCAG.org Security Posture

How UAIU Holdings Corp (operator of ADAWCAG.org) protects authentication, tenant data, encryption keys, audit trails, and infrastructure for federal subcontracts and enterprise customers. Last updated July 22, 2026.

1. Overview

ADAWCAG.org is a multi-tenant SaaS platform delivering ADA / WCAG accessibility scanning, remediation, and audit deliverables for federal prime contractors, state and local agencies, and enterprise buyers. Our platform processes URLs belonging to customer organizations and stores accessibility findings keyed to each tenant. The controls below are the day-to-day baseline applied across the product.

This document is a public posture statement, not a third-party report. SOC 2 and ISO 27001 audits are tracked separately; contact justin@adawcag.org for current attestation status.

2. Authentication

Sessions are issued as a signed JSON Web Token (HS256) using the `jose` library and stored in the `app_session` cookie. The cookie is `HttpOnly`, `SameSite=Lax`, and marked `Secure` in production. Tokens carry only the minimum claims required for routing (`userId`, `tenantId`) and are signed with `JWT_SECRET`, which is sourced from the platform secrets store and never committed.

Lifetime & rotation

- Session expiration: 7 days from issue. Re-authentication issues a fresh token.
- Logout invalidates the cookie immediately by setting `maxAge=0`.
- Server-side verification re-checks the JWT signature and expiry on every request.

Password storage

Passwords are hashed with `bcryptjs` at cost factor 12 before storage. Plaintext passwords are never logged, never written to disk, and never returned from any API. Password comparisons are performed in constant time via `bcrypt.compare`.

CSRF protection

Authentication endpoints (login, signup) and other sensitive state-mutating routes require a matching CSRF token. A 32-byte random token is issued in the `csrf_token` cookie at first contact and must be echoed back in the `x-csrf-token` header on protected POST requests, where the server compares the header value against the cookie value (double-submit pattern).

3. Authorization

Each user has exactly one role within their tenant: `OWNER`, `ADMIN`, or `MEMBER`. Roles gate write access at the API layer.

- **OWNER** — billing, plan changes, member removal, tenant deletion.
- **ADMIN** — invites, RFP and procurement records, scan configuration, integrations.
- **MEMBER** — run scans, view dashboards, edit issues assigned to them.

4. Data Protection

Encryption at rest

Sensitive secrets (SSO client secrets, OAuth refresh tokens, outbound webhook signing keys) are wrapped with AES-256-GCM envelope encryption before being written to PostgreSQL. The implementation in `lib/crypto.ts` generates a fresh 96-bit IV per record, captures the 128-bit auth tag, and stores the ciphertext as `enc:<iv>:<tag>:<data>`. The 32-byte master key (`MASTER_KEY_BASE64`) is required in production — startup refuses to encrypt without it, and decrypt rejects malformed or wrong-length authentication tags.

Encryption in transit

- HTTPS only on the public surface; HSTS and secure cookies in production.
- PostgreSQL connections use TLS to the managed database provider.
- Outbound integrations (Calendly, ZeptoMail, S3/R2) all use HTTPS.

Secret management

All credentials (`JWT_SECRET`, `MASTER_KEY_BASE64`, `ANTHROPIC_API_KEY`, `ZEPTOMAIL_TOKEN`, `GOOGLE_PLACES_API_KEY`, SMTP credentials, and database URLs) are injected at runtime from the platform secrets store. They are not present in the repository, in build artifacts, or in client-side bundles.

5. Tenant Isolation

Every tenant-owned record carries a `tenantId` column, and all Prisma queries that read or write tenant data are scoped by that column. Sessions resolve to a single `tenantId` on every request via `requireSessionContext()`, and downstream queries pass that ID into `where` clauses so a user can never read across tenant boundaries.

Agency portals

Government and agency partners receive scoped `PortalTenant` rows that bind a public-facing slug to both the operating agency tenant and the underlying client tenant. Host-based middleware

6. Audit Logging & Retention

Two append-only tables capture privileged activity:

- `AuditLog` — per-tenant record of authentication events, configuration changes, and scheduled jobs (e.g. retention purges). Rows include actor email, action verb, target ID, source IP, user agent, and a JSON `meta` blob.
- `AuditorAction` — captures human auditor activity (verifications, remediation sign-off) for evidence chains required by federal procurement.

Retention

Each tenant has a configurable retention window stored on the `Tenant` row (default 365 days). The `worker/retention.ts` job runs on a schedule, deletes scans (and their dependent issues, pages, summaries, and assets) older than the cutoff, and writes a `RETENTION_PURGE_COMPLETED` entry into the audit log so the deletion itself is auditable.

7. Infrastructure

- **Application** — Next.js 15 (App Router) on Node.js, deployed via Replit Deployments behind mTLS-proxied HTTPS.
- **Database** — managed PostgreSQL accessed via Prisma; connection-pooled and TLS-protected.
- **Queue** — Redis + BullMQ for scan jobs, with an in-process polling fallback if Redis is unreachable so scans never silently fail.
- **Object storage** — S3-compatible (R2) buckets for scan evidence, screenshots, and exported PDFs.
- **Browser automation** — Playwright with a pinned Chromium build for axe-core, pa11y, Lighthouse, and PDF generation.
- **Transactional email** — ZeptoMail with SPF/DKIM-aligned sender domain.
- **AI inference** — Anthropic Claude API for sales pitch assistance and quiz analysis; no customer scan data is sent to model providers.

8. SSO & SCIM

Enterprise plans support OIDC and SAML 2.0 single sign-on. Identity provider client secrets and SAML signing certificates are stored using the same AES-256-GCM envelope encryption described above and decrypted only inside the request-handling process.

SCIM 2.0 user provisioning is available so customers can create users from their identity provider via the `/api/scim/v2/Users` endpoint, gated by a tenant-scoped bearer token (`SCIM_BEARER_TOKEN`). Update, deactivation, and just-in-time deprovisioning are on the enterprise roadmap; contact us for current status.

9. Webhook Security

Outbound webhooks — every payload is signed with an HMAC over the request body using a per-tenant secret. Receivers verify the signature against the `X-ADAWCAG-Signature` header before trusting the payload. Secrets are rotatable from the dashboard; rotation re-encrypts and re-issues without downtime.

Inbound webhooks — the Calendly integration verifies the signing key on each event before any database write. Unsigned or signature-mismatched events are rejected with HTTP 401.

10. Vulnerability Management & Incident Response

Disclosure

Please report suspected security issues to security@adawcag.org. We acknowledge reports within 1 U.S. business day and provide an initial triage assessment within 5 U.S. business days. Researchers acting in good faith and within scope will not face legal action from UAIU Holdings Corp.

Triage targets

- **Critical** (RCE, auth bypass, cross-tenant data exposure) — patch within 72 hours of confirmation.

11. Contact

Security disclosures: security@adawcag.org

Procurement & sales: justin@adawcag.org

Customer support: support@adawcag.org

Phone: 307-414-2929

Federal identifiers: UAIU Holdings Corp · CAGE 1AUK4 · UEI WNZJXHNPC2K3 · SAM.gov Active