SHADOW AI
DISCOVERY · SECSTUDIO

SECURITY ASSESSMENT · DAST

Unauthenticated Security Assessment

DAST review of the Shadow AI Discovery web estate — findings, remediation, and post-fix verification.

AUDIENCE	VERSION	DATE
Security · Engineering · CISO	1.0	14 August 2026

Executive summary

This is an **unauthenticated, black-box dynamic assessment (DAST)** of the Shadow AI Discovery public web estate: the marketing site (`shadow-ai-discovery.com`, `www`), the articles site (`articles.shadow-ai-discovery.com`), and the product dashboard (`app.shadow-ai-discovery.com`). It was performed from the public internet with no credentials, to answer one question directly: **what can an anonymous attacker see and reach?**

Headline result: no unauthenticated access to customer data or dashboard functionality was found — every data and administrative API endpoint correctly enforces authentication. The issues that were found are **surface-reduction and hardening gaps**: the dashboard exposed its full API schema, deploy-time backup folders were reachable, and standard security response headers were absent. **All findings were remediated and re-tested; every fix is confirmed live.**

Findings at a glance

#	FINDING	SEVERITY	STATUS
1	Public API documentation & OpenAPI schema on the dashboard	Medium	Fixed
2	Deploy backup directories served from the web root	Medium	Fixed
3	Missing HTTP Strict Transport Security (HSTS)	Medium	Fixed
4	Missing clickjacking protection (X-Frame-Options / frame-ancestors)	Medium	Fixed
5	Missing Content-Security-Policy	Low	Fixed
6	Missing hardening headers on the static sites	Low	Fixed
7	Catch-all returned 200 for non-existent paths	Low	Fixed
8	Web-server product disclosure via `Server` header	Info	Fixed

Scope & method

- **Targets:** `shadow-ai-discovery.com`, `www.shadow-ai-discovery.com`, `articles.shadow-ai-discovery.com`, `app.shadow-ai-discovery.com`.
- **Perspective:** external, unauthenticated (no login, no API key).
- **Technique:** manual DAST — TLS and security-header inspection, HTTP-method testing, endpoint and API-surface enumeration, authentication-bypass probing on every discovered endpoint, sensitive-file and backup-path checks, CORS and directory-listing tests. Non-destructive; no exploitation of live data.
- **On the reported concern** ("the dashboard is reachable without authentication"): confirmed and clarified below — the application *loads* anonymously (it is a web app), but all data and actions are authentication-gated. The genuine exposure was the API *schema*, now removed.

What was verified secure (no action needed)

These were tested and behaved correctly:

- **Every data/admin API endpoint enforces authentication.** `GET /api/devices`, `/api/risks`, `/api/findings`, `/api/leads`, `/api/users`, and all others returned **401** unauthenticated. No customer or fleet data is reachable anonymously.
- **Sensor endpoints require credentials.** `POST /api/enroll`, `/api/ingest`, `/api/heartbeat` returned **401** without the enrollment secret / device token.
- **No real sensitive files are exposed.** Probes for `.git/config`, `app.env`, source files and the database returned the site index, not the file — no source or secret leakage.
- **CORS is correctly scoped.** The public search-assist endpoint rejects untrusted origins (no `Access-Control-Allow-Origin` for `evil.com`).
- **Transport:** valid Let's Encrypt certificates; HTTP redirects to HTTPS (308); `TRACE` disabled (405).
- **Sessions** use bearer tokens in `sessionStorage`, not cookies — no cookie-flag exposure — and are cleared on tab close.

Findings & remediation

1 — Public API documentation & OpenAPI schema · Medium · Fixed

The dashboard exposed FastAPI's interactive docs and machine-readable schema unauthenticated: `GET /openapi.json` (26 KB) enumerated the **entire API surface** — ~40 endpoints including `/api/users`, `/api/ingest`, `/api/dlp/inspect`, `/api/directory/sync` — alongside `/docs` and `/redoc`. This hands an attacker a complete map of the authenticated attack surface.

Fix (verified): the application now runs with `docs_url`, `redoc_url` and `openapi_url` disabled. `/docs`, `/redoc` and `/openapi.json` return **404**.

2 — Deploy backup directories served from the web root · Medium · Fixed

Data-preserving deployments had written timestamped backup folders (`.bak-*`) **inside** the web roots (`/opt/shadow-ai-site`, `/opt/shadow-ai-articles`). Because the web server serves those roots, old copies of pages were directly retrievable (e.g. `/.bak-bnd/index.html`) — stale-content and information disclosure.

Fix (verified): all `.bak-*` directories were removed from the web roots and now return **404**. The deployment process stores backups outside the served root going forward.

3 — Missing HTTP Strict Transport Security · Medium · Fixed

No `Strict-Transport-Security` header was present on any host, leaving first-request downgrade / SSL-strip exposure despite the HTTPS redirect.

Fix (verified): `Strict-Transport-Security: max-age=31536000; includeSubDomains` (with `preload` on the static edge) is now returned on all hosts.

4 — Missing clickjacking protection · Medium · Fixed

No `X-Frame-Options` or CSP `frame-ancestors`, so the login page and pages could be framed by a malicious site (clickjacking / UI-redress).

Fix (verified): `X-Frame-Options: DENY` and CSP `frame-ancestors 'none'` are now set everywhere.

5 — Missing Content-Security-Policy · Low · Fixed

No CSP meant no defence-in-depth against injected script/content.

Fix (verified): a scoped CSP is applied on every host — `default-src 'self'`, framing denied, `object-src 'none'`, `base-uri 'self'`, with the exact allowances each site needs (self-hosted assets; the fonts CDN and the dashboard API origin for the articles search) and nothing more.

6 — Missing hardening headers on the static sites · Low · Fixed

The marketing and articles sites lacked `X-Content-Type-Options`, `Referrer-Policy` and `Permissions-Policy`.

Fix (verified): `X-Content-Type-Options: nosniff`, `Referrer-Policy: no-referrer`, and a restrictive `Permissions-Policy` are now returned on all hosts.

7 — Catch-all returned 200 for non-existent paths · Low · Fixed

The static sites fell back to serving the index page for unknown paths, so probes for sensitive-looking files returned `200` — noisy for scanners and masking of true 404s.

Fix (verified): the index fallback was removed; unknown paths now correctly return **404** while clean URLs and the homepage continue to resolve.

8 — Web-server product disclosure · Info · Fixed

Responses advertised `Server: Caddy` / `Server: uvicorn`.

Fix (verified): the `Server` header is now stripped at the edge on all hosts.

Post-fix verification

All fixes were re-tested from the public internet after deployment:

- Security headers (HSTS, CSP, X-Frame-Options, X-Content-Type-Options, Referrer-Policy, Permissions-Policy) — **present on all hosts**; `Server` header **hidden**.
- `/docs`, `/openapi.json`, `/redoc` — **404**.
- `/.bak-*/…`, `/.git/config`, `/app.env`, source files — **404**.
- Regression: homepage, `/security.html`, `/kb/`, the articles index, an article page, `search-index.json`, and the dashboard SPA — **all 200**; the articles AI search still returns results; every data API still returns **401**.

Residual risk & optional hardening

The estate is in a materially stronger position; the remaining items are optional, risk-based choices rather than defects:

- **Dashboard reachability.** The dashboard *application* is internet-reachable by design (users sign in; the endpoint sensor posts to `/api/ingest`). If policy requires it to be invisible to anonymous users entirely, options are an **IP allow-list / VPN** for the console with a **narrow, separately-authenticated ingest path** for sensors, or **mutual TLS** for sensor traffic. This is an architecture decision, not a bug.
- **CSP tightening.** The current policy allows `'unsafe-inline'` because the front-ends use inline handlers/styles; moving to nonces/hashes would remove that allowance over time.
- **WAF / rate-limiting at the edge** for the public form and search endpoints (the search-assist endpoint already has per-IP rate limiting and a hard daily budget cap).

Conclusion

No unauthenticated access to data or functionality was found. Eight surface-reduction and header-hardening findings were identified, remediated, and confirmed fixed in the same engagement. The public web estate now presents a minimal, well-hardened footprint to an anonymous attacker.