// SECURITY_POSTURE_PACK

DefDossier

Tender-Ready Evidence for Defence Cyber Teams

This document summarises the security and privacy controls implemented by DefDossier as of the date below. It is intended to support enterprise procurement review and complete vendor security questionnaires for prospective customers.

This pack is not a formal audit report. Where formal third-party attestation (SOC 2, ISO 27001) is planned but not yet completed, it is disclosed explicitly in the Formal Audit section below. Any control marked “IMPLEMENTED” is verifiable directly in the DefDossier codebase or runtime configuration.

TOTAL CONTROLS	IMPLEMENTED	IN PROGRESS	PLANNED	COVERAGE
18	13	2	3	72%

// AS_OF · 2026-06-21

// 01_ARCHITECTURE

Application architecture

DefDossier is a Python (FastAPI) + React single-tenant-per-org multi-tenant SaaS. The Python API is stateless behind the Cloudflare edge; persistent state lives in a managed MongoDB cluster operated by the Emergent platform. Anthropic Claude Sonnet 4.5 provides the Sentinel AI assistant and grades practical exercises. Transactional email is delivered via Resend over Amazon SES (Tokyo region).

Tenant isolation is enforced in code: every user-generated document carries an `org_id` field stamped on write. Every cross-user read is filtered by the requester's `org_id`. There is no shared mutable state between orgs at the application layer.

// 02_CONTROLS_INVENTORY

Implemented controls by domain

Access Control

AC-01 Password storage hashing — IMPLEMENTED

All passwords are stored as bcrypt hashes (cost factor 12) with per-record salts. No plaintext or reversible encoding is ever persisted.

`backend/core.py — hash_pw(), verify_pw()`

AC-02 Brute-force protection — IMPLEMENTED

Per IP+email login attempts are tracked. After 5 failures within 15 minutes the account is locked until the cooldown elapses.

`backend/routers/auth.py — login() lockout logic`

AC-03 Session security (httpOnly + Secure + SameSite cookies) — IMPLEMENTED

Access and refresh tokens are issued as httpOnly cookies with Secure and SameSite=None attributes. Cookie bodies never traverse JavaScript.

`backend/core.py — set_cookies()`

AC-04 CSRF protection (double-submit cookie pattern) — IMPLEMENTED

Every state-changing request requires a header (X-CSRF-Token) matching a per-session cookie. Five public bootstrap endpoints are exempt by design.

`backend/core.py — CSRFMiddleware`

AC-05 Token revocation on credential change — IMPLEMENTED

User token_version increments on password reset. All previously-issued JWTs are invalidated server-side because the validator compares ver to user.token_version.

`backend/core.py — get_current_user(); backend/routers/auth.py — reset_password()`

AC-06 Password reset link expiry — IMPLEMENTED

Reset tokens are single-use, expire in 15 minutes, and rate-limit at 3 requests per email per hour. Tokens are also invalidated by any subsequent reset request.

`backend/routers/auth.py — forgot_password(), reset_password()`

Data Isolation

DI-01 Per-tenant org_id scoping — IMPLEMENTED

Every user-generated document (enrollments, certifications, lesson_progress, practical_attempts, audit_log, chat_messages) is stamped with org_id on write. Cross-org queries are blocked at the route layer (squad, suggest, plan, MITRE coverage, compliance pack).

`backend/routers/*.py — _trainee_in_org() pattern`

DI-02 Verifiable badge signatures — IMPLEMENTED

Every Open Badge certificate carries a signed JWT (HMAC-SHA256) that any third party can verify at the public /verify/{badge_id} endpoint. Tampered or revoked badges return INVALID.

`backend/badges.py — verify_badge_jwt()`

Encryption

EN-01 Encryption in transit (TLS 1.3) — **IMPLEMENTED**

All HTTP traffic to defdossier.com is terminated at Cloudflare with TLS 1.3 and HSTS. The Emergent edge does not accept plaintext HTTP for the production hostname.

`Cloudflare DNS + Emergent edge config`

EN-02 Encryption at rest — **IN PROGRESS**

Production database is MongoDB hosted by the Emergent platform. Volume-level encryption at rest is dependent on the underlying managed-Mongo offering; awaiting written attestation from Emergent infrastructure.

`Infrastructure-level (out of repo scope)`

Audit & Monitoring

AU-01 Authentication audit log — **IMPLEMENTED**

Password reset requests and completions, along with compliance pack email sends, are appended to db.audit_log with user_id, IP, and timestamp. Records are append-only from the application.

`backend/routers/auth.py, backend/routers/compliance_routes.py — db.audit_log.insert_one()`

AU-02 Compliance pack delivery audit trail — **IMPLEMENTED**

Every compliance pack email send records the sender, recipient, subject user, and Resend message id — enabling later proof of delivery to procurement officers.

`backend/routers/compliance_routes.py`

Email Security

EM-01 SPF + DKIM + DMARC on sending domain — **IMPLEMENTED**

defdossier.com publishes SPF (include:amazonses.com), DKIM (Resend resend._domainkey), and DMARC (p=none, monitoring) records. Verified by Resend domain authentication.

`Cloudflare DNS records`

Vendor Management

VM-01 Documented subprocessors — **IMPLEMENTED**

Customer data is processed only by the subprocessors listed in the Trust Center. Any addition triggers a public update.

`frontend/src/pages/Trust.jsx — Subprocessors section`

Formal Audit

PG-01 SOC 2 Type I attestation — **PLANNED**

Targeted 2027. Pre-audit controls implementation underway; engagement with an independent CPA partner expected once first paid customers are onboard.

`Roadmap — not yet in scope`

PG-02 ISO 27001 certification — **PLANNED**

Under evaluation. ISO 27001 will be pursued if customer demand and revenue justify the audit cost.

`Roadmap — not yet in scope`

PG-03 External penetration test — **PLANNED**

Targeted Q3 2026 once seat count exceeds 50. Will engage a CREST/OSCP-credentialed firm.

Roadmap — not yet in scope

Business Continuity

PG-04 Disaster-recovery and backup policy **IN PROGRESS**

Daily managed-Mongo snapshots via Emergent platform. RTO/RPO targets being formalised.

Infrastructure — Emergent managed Mongo

// 03_SUBPROCESSORS

Data subprocessors

Customer data is processed by the third parties listed below. Each addition or change to this list will be reflected in the public Trust Center within 30 days.

Vendor	Purpose	Data category	Region
Cloudflare	DNS, TLS termination, edge CDN	Encrypted traffic only	Global edge
Emergent	Application hosting, managed MongoDB	All application data	Multi-region
Resend	Transactional email delivery	Recipient email, message body	Tokyo (ap-northeast-1)
Amazon SES (via Resend)	SMTP relay	Email envelope + body	AP Northeast 1
Anthropic	Sentinel AI assistant	Prompt + chat context for an active session	United States

// 04_INCIDENT_RESPONSE

Incident response

DefDossier maintains a basic incident response process appropriate to its current stage: (1) detection via application logs and Resend bounce monitoring, (2) containment by token-version invalidation and key rotation, (3) notification to affected customer org_admins within 72 hours of a confirmed security incident, (4) post-incident review captured in writing. A formal IR runbook is targeted alongside SOC 2 Type I preparation.

// 05_DATA_LIFECYCLE

Data retention and deletion

Customer data is retained for the lifetime of the subscription plus 30 days. On written deletion request from the org_admin, DefDossier will purge all org-scoped records (users, enrollments, certifications, lesson progress, practical attempts, audit log, chat history) within 30 days. Verifiable badges remain individually verifiable post-deletion only at the badge holder's discretion.