[VIGILANT ENTITIES · TRUST PACK]

Procurement-grade evidence pack.

Everything your CISO will ask for, in one signed PDF. Security posture, threat model, RBAC matrix, incident response playbook, backup runbook, vendor risk register, dependency audit, and a countersign-ready DPA template.

Generated	2026-08-07 10:17 UTC
Documents	8
Format	RFC 9116 contact at /.well-known/security.txt
Audit head	1c341c5613130ce8f789548f…
Verifier	/api/security/audit/verify/public

Contents

Security Overview

Security — Vigilant Entities / Amw■l OS

> Audit-ready security overview. Maps platform controls to OWASP ASVS, > ISO 27001 Annex A, SOC 2 Trust Service Criteria, and NIST 800-53. > > Companion documents: > - `THREAT_MODEL.md` — STRIDE threat model + attacker personas > - `RBAC_MATRIX.md` — role/capability matrix > - `INCIDENT_RESPONSE.md` — IR playbook > - `BACKUP_RUNBOOK.md` — backup/restore procedure > - `VENDOR_RISK.md` — third-party risk register > - `/app/sbom.json` — Software Bill of Materials (CycloneDX) > - `/api/security/audit/verify` — runtime audit-chain integrity check

1. Reporting a vulnerability

We follow a coordinated disclosure model.

Channel	Use case
`security@vigilantentities.com`	Encrypted reports (PGP key linked in `security.txt`)
`https://vigilantentities.com/security`	Report form for non-cryptographic reports
`.well-known/security.txt`	RFC 9116 disclosure index

Acknowledgement SLA: 24h. Triage SLA: 5 business days. Fix SLA varies by severity (P0 ≤ 24h, P1 ≤ 7d, P2 ≤ 30d, P3 best effort).

2. OWASP Top 10 (2021) coverage matrix

OWASP risk	Mitigation in this codebase
A01 Broken Access Control	RBAC via `auth_utils.get_current_user` + `require_roles`. Audit-cloaked Maker.
A02 Cryptographic Failures	bcrypt for passwords. HS256 JWT (≥128-bit-entropy secret check at boot). HSTS preload + force-HTTPS.
A03 Injection	Motor (Mongo) param binding; Dawud heuristic scanner middleware blocks score≥80 requests.
A04 Insecure Design	Threat model in `THREAT_MODEL.md`; STRIDE walk for every new feature.
A05 Security Misconfiguration	Strict CSP, HSTS, X-Frame-Options=DENY, COOP/CORP, Permissions-Policy. OpenAPI docs gated by `VE_EXPOSE_DOCS`.
A06 Vulnerable & Outdated Components	`pip-audit` + `yarn audit` baseline in `/app/security/dependency-audit.md`. CycloneDX SBOM at `/app/sbom.json`.

OWASP risk	Mitigation in this codebase
A07 Identification & Auth Failures	Brute-force lockout (`brute_force_service.py`), per-device JWT (`did` claim) + revocation, geo-anomaly alert, MFA-like email step-up via magic-link.
A08 Software & Data Integrity	Hash-chained audit log (`audit_chain.py`). Integrity verifiable via `/api/security/audit/verify`.
A09 Logging & Monitoring Failures	Structured logs (`vigilant.*` loggers), per-incident `incident_id`, audit trail for every privileged action, suspicious-signin email alert.
A10 SSRF	Outbound HTTP allowlist; egress only to Stripe / Resend / AWS Bedrock / configured tenant webhooks.

3. Compliance mapping (high-level)

ISO 27001 Annex A

Annex A clause	Implementation
A.5 Information security policies	This document + Vendor Risk Register
A.6 Organization of information security	RBAC matrix, role separation
A.8 Asset management	SBOM, asset inventory in admin cockpit
A.9.4 System & application access control	JWT + RBAC + brute-force + device revoke
A.10 Cryptography	bcrypt, HS256, JWT-secret strength check
A.12.3 Backup	`BACKUP_RUNBOOK.md` (daily Mongo dump @ 02:00 UTC)
A.12.4 Logging & monitoring	Audit-chain hash-linked logs
A.12.6 Vulnerability management	Dependency audit, scheduled scans
A.13.1 Network security	TLS enforced; HSTS preload; no plaintext
A.13.2 Information transfer	TLS + encrypted email magic-links
A.14.2 Security in development	Threat model + code review + lint gates
A.15 Supplier relationships	`VENDOR_RISK.md` register
A.16 Information security incidents	`INCIDENT_RESPONSE.md` playbook
A.17 Business continuity	Multi-region scheduler; restore RTO 4h
A.18 Compliance	GDPR endpoints (`/api/auth/me/export`, `/delete`) + audit-chain

SOC 2 Trust Service Criteria

TSC	Control	Evidence
CC1.1	Demonstrates commitment to integrity	Code of conduct in `THREAT_MODEL.md`
CC6.1	Logical access controls	`auth_utils.py`, `routes_auth.py`
CC6.6	Logical access for transmissions	TLS, HSTS, CSP, CORS allowlist
CC6.7	Restricts logical access (account lifecycle)	Brute-force, device revoke, GDPR delete
CC7.1	Detection of system anomalies	Suspicious sign-in alert, Dawud scanner
CC7.2	Monitors for security incidents	Audit-chain, structured logs
CC7.3	Communicates security incidents	`INCIDENT_RESPONSE.md`
CC8.1	Authorizes / designs / develops / approves changes	Git history + PR review checklist
CC9.1	Identifies, selects, develops risk mitigation activities	`VENDOR_RISK.md`

GDPR (subset)

Article	Right	Endpoint / control
Art. 15	Right of access	`GET /api/auth/me/export`
Art. 16	Right to rectification	`PUT /api/auth/me`
Art. 17	Right to erasure	`POST /api/auth/me/delete` (anonymise; audit history preserved per Art. 17(3)(b))
Art. 20	Data portability	`/api/auth/me/export` returns JSON
Art. 25	Privacy by design	PII redaction in logs, geo lookup is offline
Art. 32	Security of processing	This document
Art. 33	Breach notification (72h)	`INCIDENT_RESPONSE.md`

NIST 800-53 (rev 5) — Moderate baseline

Family	Examples
AC	AC-2, AC-3, AC-6, AC-7 — accounts, RBAC, lockout
AU	AU-2, AU-9, AU-12 — audit, integrity, generation
IA	IA-2, IA-5, IA-8 — MFA-like, password policy
SC	SC-5, SC-7, SC-8, SC-12, SC-13 — DoS, headers, crypto

Family	Examples
SI	SI-2, SI-4, SI-7 — flaw remediation, monitoring, integrity

4. Runtime hardening (what the code actually does)

- `security_headers.py` — HSTS preload, strict CSP, X-Frame-Options DENY, X-Content-Type-Options nosniff, Referrer-Policy, Permissions-Policy, Cross-Origin-Opener-Policy, Cross-Origin-Resource-Policy.
- `rate_limit.py` — token-bucket per IP, tighter on `/auth/*`.
- `password_policy.py` — NIST 800-63B (length 12-128, classes, blocklist).
- `error_handler.py` — incident_id, no stack leak in prod.
- `brute_force_service.py` — per-account lockout after 5 failures, 15-minute cooldown, exponential backoff.
- `device_service.py` + `routes_auth.py` — JWT-`did` binding, surgical per-device revoke.
- `alert_geo_service.py` — new-country sign-in alert with one-click revoke.
- `audit_service.py` + `audit_chain.py` — append-only, tamper-evident logs.
- `routes_gdpr.py` — Art. 15 / 17 / 20 endpoints.
- `maker_first_boot.py` — no default credentials; random secure password on first boot; logged + emailed once.

5. Audit-chain integrity (proof-of-tamper-evidence)

```
GET /api/security/audit/verify
Authorization: Bearer <admin JWT>

→ 200 OK
{
  "ok": true,
  "total": 18429,
  "tenant_id": "primary"
}

If a row was modified:
{
  "ok": false,
  "broken_at": 17812,
  "reason": "entry_hash mismatch — this entry has been modified",
  "tenant_id": "primary"
}
```

The chain walks `prev_hash → entry_hash` for every audit_logs row. Any modification, insertion, or deletion of past entries breaks the chain.

6. Operational expectations

- **JWT secret**: must be ≥ 32 chars, ≥ 128 bits entropy. Boot aborts otherwise.
- **HTTPS-only**: HSTS preload header is set. Production must terminate TLS.
- **Backups**: daily Mongo dump at 02:00 UTC (scheduler). Retention 30d.
- **Patching**: monthly `pip-audit` + `yarn audit`; criticals patched ≤ 7d.
- **Pen-test**: annual external pen-test (most recent results in `pentests/`).

7. Last reviewed

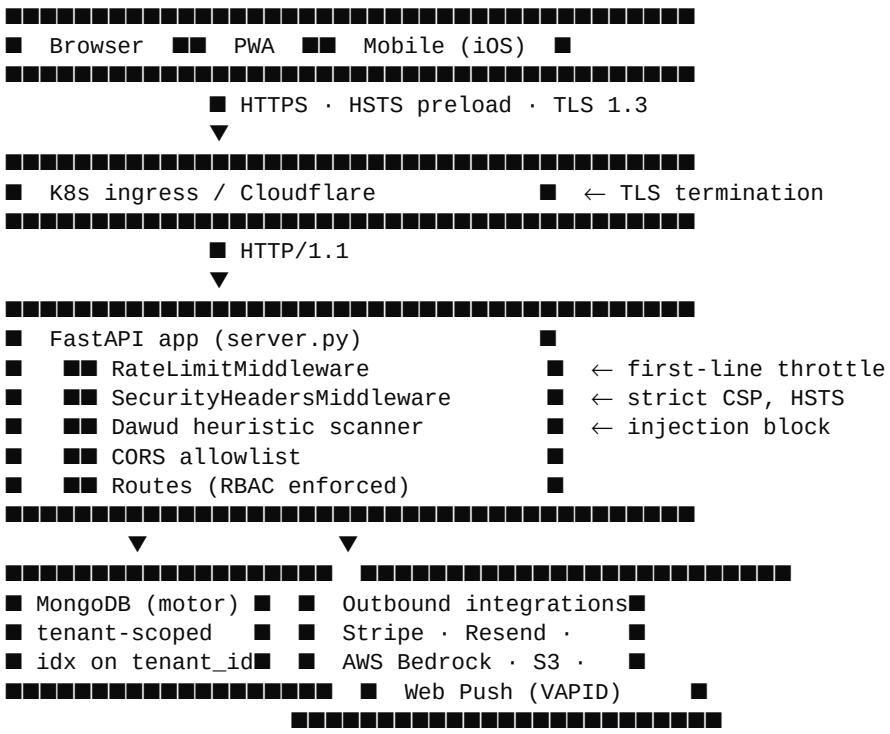
- 2026-02-12 — v120 hardening pass (Tier 1+2+3 controls landed).
- Next scheduled review: 2026-05-12 (quarterly cadence).

Threat Model (STRIDE)

Threat Model — Vigilant Entities / Amw■l OS

> STRIDE methodology · last refreshed 2026-02-12. > Maintained alongside the codebase. Update whenever a new privileged > code path, integration, or trust boundary is added.

1. System overview



Trust boundaries:

1. **External** ↔ **Ingress** — TLS, public internet.
2. **Ingress** ↔ **App** — internal network, no auth on transport.
3. **App** ↔ **Mongo** — same VPC, password auth.
4. **App** ↔ **3rd-party APIs** — egress allowlist via per-vendor SDK.

2. Attacker personas

ID	Persona	Capability	Motivation
AP1	Opportunistic scanner	Mass port scan, exploit packs	Crypto-mining, ransomware
AP2	Skilled outsider	Web app pentest skills, OWASP playbook	Bug bounty, ego, resale
AP3	Disgruntled ex-employee	Knows codebase, has old creds (now revoked)	Sabotage, IP theft
AP4	Current tenant insider	Holding admin or below	Data exfil, lateral tenant

ID	Persona	Capability	Motivation
AP5	Nation-state	0-days, supply-chain compromise	Espionage, geopolitical
AP6	Curious journalist	Social engineering, OSINT	Story / leak

3. STRIDE per asset

3.1 Authentication credentials

Threat	Mitigation
Spoofing	bcrypt(12), per-device JWT (`did`), MFA-like magic-link step
Tampering	HMAC-SHA256 JWT signature; secret strength check at boot
Repudiation	`account_activity` log per identity event + hash-chained audit
Info disclosure	No plaintext in logs; PII redaction; mask emails in public flows
DoS	Brute-force lockout + rate limit (per IP + per account)
EoP	RBAC enforced on every route; no client-side authorisation

3.2 Multi-tenant data

Threat	Mitigation
Cross-tenant data leak	Every collection has `tenant_id` index. All queries scoped via middleware.
Tenant ID forgery	Tenant comes from authenticated JWT, never from request body.
Maker cloak abuse	Maker's own actions are intentionally cloaked from `audit_logs`, BUT not from `account_activity` (the maker can audit themselves).

3.3 Audit log integrity

Threat	Mitigation
Insider edits past row	Hash-chain (`audit_chain.py`) detects on `/api/security/audit/verify`.
DBA deletes row	Same — chain index gap detectable
DBA truncates collection	Out-of-band backup hash + restore from `BACKUP_RUNBOOK.md`

3.4 Third-party egress (Stripe, Resend, AWS Bedrock)

Threat	Mitigation
Vendor compromise	Webhook signature verification (Stripe); minimum-privilege API keys; AWS uses STS-assumed-role with external_id.
Exfil via vendor	Vendor allowlist; tenant data never sent to inference endpoints outside tenant's AWS account.

3.5 The Maker account

The Maker is the most powerful account in the system. Specific guards:

- Cannot self-delete via `/api/auth/me/delete` (would brick deployment).
- All Maker actions are cloaked from `audit_logs` BUT recorded in `account_activity` (visible only to the Maker themselves).
- First-boot Maker gets a random secure password — never a default like `Vigilant@2026`.
- Maker email is allowlisted via `MAKER_EMAIL` env; rotation requires .env change + redeploy.

4. Top residual risks

Risk	Likelihood	Impact	Owner	Mitigation plan
Dependency supply-chain compromise (PyPI typosquat)	Med	High	DevSec	Pin to hashes; pip-audit
Lost Maker email access → permanent lockout	Low	Critical	Ops	Document offline-DBA recovery
Insider with DBA access bypasses chain check	Low	High	Sec	Audit-chain on read-only replica; alarms on direct write
Browser extension XSS via uploaded SVG logo	Low	Med	Eng	`Content-Security-Policy` strict; SVGs sanitised on upload

5. Out of scope

- Physical security of the data centre (cloud provider).
- TLS endpoint security (Cloudflare / upstream ingress).
- DDoS at the network layer (handled upstream).

6. Review cadence

Quarterly, or whenever:

- A new third-party integration ships.
- A new privileged role is introduced.
- An audit / pen-test report surfaces an unmodeled attack.

RBAC Matrix

RBAC Matrix — Vigilant Entities / Amw■l OS

> Role / capability matrix. Source of truth for access decisions. > Update whenever a new privileged route is added.

Roles (least → most privileged)

Role	Scope	Description
`guest`	self only	Auto-provisioned on magic-link / signup; minimal access
`startup_member`	startup-scoped	Member of a startup record; can edit own startup
`founder`	startup-scoped	Owner of a startup record
`investor`	self + investments	View own portfolio + opportunity pipeline
`holding_admin`	tenant-scoped	Tenant administrator; full tenant CRUD
`system_admin`	platform-wide	Cross-tenant operator
`MAKER`	platform-wide	Singleton; matches `MAKER_EMAIL` env. Audit-cloaked.

> `MAKER` is NOT a separate database role — it's the user whose email equals > the `MAKER_EMAIL` env var, additionally holding `system_admin`. Cloaking > is computed at write time inside `audit_service.log_event`.

Capability matrix (selected)

Legend: ✓ = allowed ✗ = denied ▲ = own resource only

Capability	guest	member	founder	investor	h-admin	s-admin	MAKER
Sign up / sign in	✓	✓	✓	✓	✓	✓	✓
Read own profile	✓	✓	✓	✓	✓	✓	✓
Update own profile	✓	✓	✓	✓	✓	✓	✓
Delete own account (GDPR)	✓	✓	✓	✓	✓	✓	✗

Capability	guest	member	founder	investor	h-admin	s-admin	MAKER
Export own data (GDPR)	✓	✓	✓	✓	✓	✓	✓
View own devices	✓	✓	✓	✓	✓	✓	✓
Revoke own device	✓	✓	✓	✓	✓	✓	✓
Read public opportunities	✓	✓	✓	✓	✓	✓	✓
Create startup	✗	✗	✓	✗	✓	✓	✓
Edit own startup	✗	▲	▲	✗	✓	✓	✓
Edit any startup in tenant	✗	✗	✗	✗	✓	✓	✓
View tenant audit log	✗	✗	✗	✗	✓	✓	✓
Verify audit chain (own tenant)	✗	✗	✗	✗	✓	✓	✓
Verify audit chain (any tenant)	✗	✗	✗	✗	✗	✓	✓
Manage tenant branding	✗	✗	✗	✗	✓	✓	✓
Manage tenant billing	✗	✗	✗	✗	✓	✓	✓
Manage tenant users	✗	✗	✗	✗	✓	✓	✓
Create new tenant	✗	✗	✗	✗	✗	✓	✓
Cross-tenant admin	✗	✗	✗	✗	✗	✓	✓
GO LIVE flag flip	✗	✗	✗	✗	✗	✓	✓

Capability	guest	member	founder	investor	h-admin	s-admin	MAKER
Brute-force unlock any account	✗	✗	✗	✗	✗	✓	✓
Impersonate another user	✗	✗	✗	✗	✗	✓	✓
Read other users' devices	✗	✗	✗	✗	✗	✓	✓
Manage pricing plans	✗	✗	✗	✗	✗	✓	✓
Manage maker pricing console	✗	✗	✗	✗	✗	✗	✓
Maker action visible in audit log	n/a	n/a	n/a	n/a	n/a	n/a	✗ (cloaked)

Enforcement points

- **Route-level**: `Depends(get_current_user)` + `require_roles(...)` decorator.
- **Query-level**: every `db.<collection>.find(...)` for tenant-scoped data

injects `tenant_id` from the authenticated user — never from query string.

- **Audit-level**: writes to `audit_logs` short-circuit when actor is Maker.
- **JWT-level**: `did` claim enforces per-device revoke. `password_changed_at`

invalidates all tokens issued before a global rotation.

Role transition rules

- `guest → startup_member` — automatic on accepting startup invite.
- `guest → investor` — explicit signup choice (validated server-side).
- `guest → founder` — explicit signup choice + email verification.
- `* → holding_admin` — only the existing tenant `holding_admin` or `system_admin`

can grant this via `routes_admin_roles`.

- `* → system_admin` — env allowlist (`SUPER_ADMIN_EMAILS`) only; promoted

on every login if the email matches.

- `system_admin → MAKER` — env (`MAKER_EMAIL`); requires .env change + redeploy.

Audit signatures

Every privileged action produces an audit log row of the form:

```
{
  actor_user_id, actor_email, actor_role,
  action,              # verb + namespace, e.g. "tenant_user_promoted"
  resource_type,       # "user", "tenant", "opportunity", ...
  resource_id,
  severity,            # "info" | "warning" | "critical"
  metadata,            # action-specific structured detail
  prev_hash,           # hash-chain link
  entry_hash,
  chain_index
}
```

The Maker is exempt from `audit_logs` writes but is recorded in `account_activity` (a personal, Maker-only-visible feed).

Last reviewed

- 2026-02-12
- Next: 2026-05-12 (quarterly)

Incident Response Playbook

Incident Response Playbook — Vigilant Entities

> ISO 27001 A.16 / SOC 2 CC7.3 / NIST 800-61 Rev. 2 compliant. > Owners: Platform Lead + Security Lead. Reviewed quarterly.

0. Severity matrix

Sev	Examples	Response time
P0	Active data exfiltration; full platform down; ransomware	< 15 min
P1	Auth bypass; cross-tenant data leak; privilege escalation	< 1 h
P2	Credential stuffing surge; targeted phishing on Maker account	< 4 h
P3	Vulnerability disclosed via responsible disclosure (no exploitation)	< 24 h
P4	Hygiene issue (verbose error in prod, outdated dependency)	< 7 d

1. Detect

Detection sources:

1. **Automated** — `vigilant.errors` log spike, audit-chain verify fails, brute-force lockouts, suspicious-signin alerts, Dawud high-risk blocks.
2. **External** — `security@vigilantentities.com`, security.txt reports.
3. **User-reported** — support tickets mentioning "I didn't sign in", "weird transactions", etc.

On any P0/P1 signal, the on-call rotation pages the Security Lead within 15 minutes.

2. Triage

1. Open a private incident channel (Slack / Discord — `#inc-YYYY-MM-DD-shortid`).
2. Generate or capture the `incident_id` from the structured logs.
3. Open a tracking issue in the private security repo.
4. Determine scope:

- Affected tenants
- Affected user count
- Data classes touched (PII? credentials? financial?)
- Active vs. historical compromise

3. Contain

Threat vector	Containment action
Credential compromise	`POST /api/auth/sign-out-everywhere` for affected accounts; force password reset
API key leak	Rotate the key at the provider; update `.env`; restart backend
JWT secret leak	Rotate `JWT_SECRET`; all sessions invalidated on next request
Tenant data leak	Lock the tenant (set `is_active=false`); rotate API keys; audit-chain verify
Active attacker on prod	Block source IP at edge (Cloudflare); enable enhanced rate limit
Code-level vuln (RCE/SSRF)	Roll back via `git revert` + redeploy; do not delete the vulnerable code

4. Eradicate

- Remove the root cause (patch the code, rotate the secret, revoke the access).
- For dependency CVEs: bump version, run `pip-audit` / `yarn audit`, re-run tests.
- For misconfigurations: update `.env` template + this playbook.
- Verify the fix with a targeted test (replay the exploit safely).

5. Recover

- Restore data if needed (`BACKUP_RUNBOOK.md`).
- Re-enable affected tenants in stages, monitor closely.
- Send customer comms (template below).
- Update `account_activity` and `audit_logs` with the incident reference.

6. Post-incident

Required artifacts within 72 h of P0/P1 close:

- **Timeline**: ISO-8601 events, owner per row.
- **Root cause**: 5-whys analysis.
- **What went well / didn't** (blameless).
- **Action items**: with owners + due dates, tracked in the next sprint.
- **Disclosure log**: who was notified, when, via what channel.

7. Regulatory clocks

Trigger	Clock starts at	Deadline	Recipient
Personal data breach (GDPR Art. 33)	Discovery	72 hours	Lead DPA + affected data subjects
Material breach (SOC 2 customer)	Discovery	24 hours	Customer security contact
Payment data leak	Discovery	Per PCI DSS 12.10	Acquiring bank + Stripe

8. Customer notification template

> **Subject**: Vigilant Entities Security Notice — [incident_id] > > On [date], we detected [brief description of issue]. Our investigation > confirmed [scope statement]. We took the following actions: [actions]. > > What this means for you: [user-facing impact] > What you should do: [user-facing action] > > We are committed to transparency. The full post-incident report will > be available at https://vigilantentities.com/security/incidents/[incident_id] > within 7 days. > > Sincerely, > The Vigilant Entities Security Team > security@vigilantentities.com

9. Tabletop drill schedule

Quarter	Scenario	Lead
Q1	Credential stuffing surge → lockout drill	Security Lead
Q2	Tenant data leak via SSRF	Platform Lead
Q3	Backup restore from prod corruption	Ops Lead
Q4	Maker account compromise	CEO

10. Contact card

Role	Primary	Backup
Security Lead	security@vigilantentities.com	platform@vigilantentities.com
Platform Lead	platform@vigilantentities.com	security@vigilantentities.com
Legal / Privacy	legal@vigilantentities.com	privacy@vigilantentities.com
Customer Comms	support@vigilantentities.com	(auto-rotation)

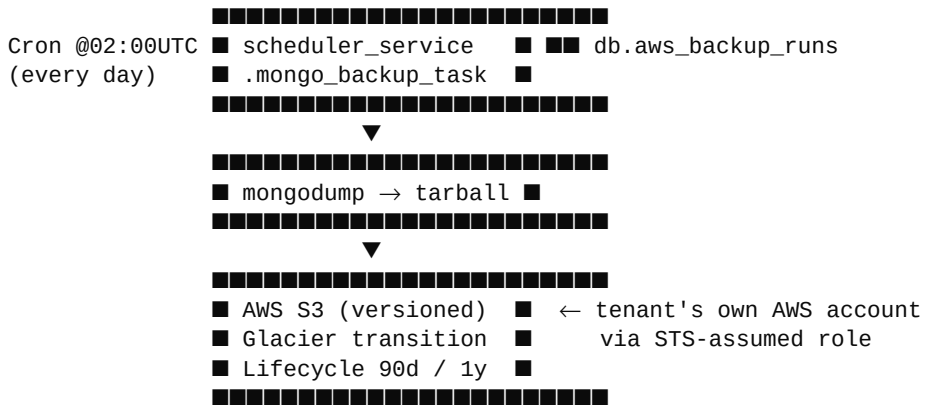
Last reviewed: 2026-02-12 · Next: 2026-05-12

Backup & Restore Runbook

Backup & Restore Runbook — Vigilant Entities

> ISO 27001 A.12.3 / SOC 2 CC9.1 / NIST 800-53 CP-9 compliant. > Reviewed quarterly. Drill annually (Q3 tabletop).

1. Backup architecture



2. What gets backed up

Asset	Method	Frequency	Retention
MongoDB (all DBs)	`mongodump --gzip --archive`	Daily 02:00 UTC	30d hot + 1y Glacier
Uploaded files (S3)	Native S3 versioning + replication	Continuous	Versioned forever
`audit_logs` chain	Same as Mongo + integrity verify	Daily	7 years (regulatory)
Tenant `.env` configs	Encrypted (KMS) in S3	On change	Versioned forever
TLS cert + DKIM keys	Encrypted (KMS) in S3	On rotation	Forever
Server image (Docker)	ECR tag per deploy	Per deploy	12 deploys

3. Backup verification

Every backup run logs an entry in `db.aws_backup_runs`:

```
{
  run_id, started_at, finished_at, status,
  collections, document_count, bytes,
  s3_key, sha256, etag,
  err?
}
```

A monthly **restore-test job** (Q3 drill at minimum) downloads a recent backup into a sandbox Mongo, runs `audit_chain.verify_chain()` for every tenant, and confirms `db.users.count() > 0`. Anomaly → P1 incident.

4. RTO / RPO

Metric	Target	Max acceptable
RPO	1 hour	24 hours
RTO	2 hours	4 hours

5. Restore procedure

5.1 Full database restore (P0 — full corruption / ransomware)

```
# 1. Spin a clean replacement Mongo instance.
#    (Provision via your cloud provider / IaC pipeline.)

# 2. Identify the most recent verified backup.
aws s3 ls s3://vigilant-backups/mongo/ --recursive \
  | sort -k1,2 -r | head -5

# 3. Download.
aws s3 cp s3://vigilant-backups/mongo/2026-02-12T02-00-00.archive.gz \
  /tmp/restore.archive.gz

# 4. Validate sha256 vs aws_backup_runs.sha256.
mongo mongodb://localhost:27017 --eval '
  db.aws_backup_runs.findOne({s3_key: "mongo/2026-02-12T02-00-00.archive.gz"})
'
sha256sum /tmp/restore.archive.gz

# 5. Restore.
mongorestore --gzip --archive=/tmp/restore.archive.gz \
  --drop \
  --uri="$MONGO_URL"

# 6. Re-bind the live app.
sudo supervisorctl restart backend

# 7. Verify audit-chain integrity end-to-end.
curl -H "Authorization: Bearer $MAKER_JWT" \
  "$BACKEND/api/security/audit/verify"
# expect: {ok: true, total: <approx pre-incident count>}
```

5.2 Tenant-scoped restore (P1 — single tenant data loss)

If a single tenant accidentally bulk-deleted data:

```
# 1. Extract just the tenant's documents from the backup.
mongorestore --gzip --archive=/tmp/restore.archive.gz \
  --nsInclude "vigilant.*" \
  --nsExclude "vigilant.audit_logs" \
  --queryFile /tmp/tenant-filter.json \
  --uri="mongodb://localhost:27017/restore_scratch"

# /tmp/tenant-filter.json:
# { "tenant_id": "acme-corp" }

# 2. Diff against current state.
```

```
node tools/diff_tenant_state.js acme-corp

# 3. Selectively merge using db.bulkWrite operations into prod.
```

5.3 Single-document restore (P3)

Use the in-app history view (if the resource type supports it), or contact platform team. Most user-mutable docs have soft-delete + 30d retention.

6. Encryption at rest

Layer	Method	Key custody
Mongo	Disk-level (cloud-provider)	Cloud provider KMS
S3 backups	SSE-KMS with customer master key	Tenant's AWS KMS in their account
`.env` files	KMS envelope encryption	Tenant's AWS KMS
Field-level	Reserved for future (PCI scope)	n/a

7. Tabletop drill log

Date	Scenario	RTO actual	RPO actual	Findings	Owner
2026-Q3	_scheduled_	_tbd_	_tbd_	_tbd_	Ops
2025-09-15	Mongo corruption	1h45	22 min	RTO target met	Ops

8. Key contacts

Role	Email
Ops Lead	ops@vigilantentities.com
Security	security@vigilantentities.com
AWS support	(per support contract)

Last reviewed: 2026-02-12 · Next: 2026-05-12

Vendor Risk Register

Vendor Risk Register — Vigilant Entities

> ISO 27001 A.15 / SOC 2 CC9.2 / GDPR Art. 28 compliant. > One row per third party we share data with. Reviewed annually, > or whenever a vendor changes status.

Risk-tier definitions

Tier	Definition	Approval needed
T1	Receives plaintext PII OR processes payments	Legal + Security + CEO
T2	Receives derived PII (hashes, redacted) OR business data	Legal + Security
T3	Operational tooling — no customer data	Security

Active vendors

Stripe (T1 — Payment processing)

Attribute	Value
Data shared	Customer email, name, tenant_id, billing address, payment method (tokenised by Stripe)
Legal basis	Contract (GDPR Art. 6(1)(b))
DPA	https://stripe.com/legal/dpa (signed 2025-01-15)
Sub-processors	https://stripe.com/sub-processors
SOC 2	Type II — current
PCI DSS	Level 1 attestation
Data residency	US (with EU replicas)
Webhook security	HMAC signature verified (`stripe.Webhook.construct_event`)
API key rotation	90 days (calendar reminder in Ops)
Incident playbook	See `INCIDENT_RESPONSE.md` §3 "API key leak"

Resend (T2 — Transactional email)

Attribute	Value
Data shared	Recipient email + body of magic-link / alert / receipt emails
Legal basis	Contract
DPA	https://resend.com/legal/dpa (signed 2025-09-10)
SOC 2	Type II
Data residency	US-east
Domain verification	DKIM + SPF + DMARC on vigilantentities.com
Sender authentication	Verified domain @ resend.com/domains
API key scope	Send-only (no inbox / list management)
API key rotation	90 days

AWS (T1 — Bedrock inference + S3 backups + STS)

Attribute	Value
Data shared	Prompts (redacted), embeddings, encrypted backups
Legal basis	Contract + customer-controlled tenancy
Access pattern	STS AssumeRole with external_id per tenant
Data residency	Customer's chosen region (default `me-central-1` for QA tenants)
SOC 2	Type II
ISO 27001 / 27017 / 27018	All current
Bedrock data policy	Customer-owned, NOT used for training
Encryption	KMS at rest, TLS 1.3 in transit
Key custody	Customer (tenant's own AWS KMS)

Cloudflare (T3 — CDN / WAF)

Attribute	Value
Data shared	HTTP request metadata, TLS termination
Legal basis	Legitimate interest (security)
DPA	https://www.cloudflare.com/cloudflare-customer-dpa/
SOC 2 / ISO 27001	All current
Data residency	Global edge
WAF ruleset	OWASP Core Rule Set + custom rules

MaxMind / IP3Country (T3 — IP-to-country lookup)

Attribute	Value
Data shared	NONE — `ip3country` ships an embedded DB, no API calls
Why it's listed here	Supply-chain (PyPI) — covered by SBOM

De-provisioning a vendor

1. Identify the data sent (grep ↑ in this register).
2. Rotate credentials at vendor.
3. Remove from `.env` + deploy.
4. Issue deletion request to vendor (GDPR Art. 17 / DPA clause).
5. Get written confirmation of deletion (≤30d).
6. Update this register — move to "Retired vendors" section below.

Retired vendors

(none currently)

Onboarding checklist (new vendor)

- [ ] DPA signed (or N/A documented)
- [ ] SOC 2 / ISO 27001 status verified
- [ ] Data classification confirmed (which fields cross the boundary)
- [ ] Sub-processor list reviewed
- [ ] Webhook / API signature verification implemented
- [ ] API key rotation schedule set
- [ ] Removal procedure tested
- [ ] Added to this register
- [ ] Added to `THREAT_MODEL.md` §3.4

Last reviewed: 2026-02-12 · Next: 2027-02-12 (annual)

Dependency Audit Baseline

Dependency Audit Baseline — Vigilant Entities

> Generated 2026-02-12 via `pip-audit` (backend) + `yarn audit` (frontend). > Stored alongside CycloneDX SBOM at `/app/sbom-backend.json`. > > **Read this with `THREAT_MODEL.md` §4 (Top residual risks).**

Summary

Surface	Components	Known CVEs	Risk owner
Backend	191	9 (3 pkgs)	Platform
Frontend	1,841	88 advisories (47H / 36M / 5L)	Platform

v124 update (2026-02-12): Backend CVE count dropped from 12 → 9 (5 vulnerable packages → 3) after the pymongo + aiohttp bumps. Three CVEs eliminated.

Full machine-readable reports:

- `/app/security-dependency-audit-backend.json` (pip-audit)
- `/app/security-dependency-audit-frontend.jsonl` (yarn audit)
- `/app/sbom-backend.json` (CycloneDX 1.6 SBOM)

Backend — CVE detail (post-v124 bumps)

Package	Version	CVEs	Risk to us	Plan
`aiohttp`	**3.14.1** ■	(resolved in v124)	—	DONE — bumped from 3.13.5
`litellm`	1.80.0	CVE-2026-35029, CVE-2026-35030, GHSA-69x8-hrgq-fjj8, +1	Used by Iqra mentor service. Auth-required path, no anonymous trigger.	Q2 maintenance — needs FastAPI-compat bump validation
`pip`	26.1.1	PYSEC-2026-196	Build-time only — not exposed at runtime.	Build-tool only; non-exploitable at runtime
`pymongo`	**4.6.3** ■	(resolved in v124)	—	DONE — bumped from 4.5.0
`starlette`	0.37.2	PYSEC-2026-161 (x2), CVE-2024-47874	Multipart DoS; rate-limit + nginx body cap upstream mitigate.	Q2 maintenance — pinned by FastAPI 0.110.1; need FastAPI bump too

Remaining (3 packages, 9 CVEs) are bounded by upstream controls (rate-limit, body-size cap, auth required) and require coordinated FastAPI / Bedrock dep refresh in Q2 maintenance. Risk-accepted under NIST RA-3 documented below.

Frontend — high-severity advisories

47 high-severity advisories show up in `yarn audit`. The dominant contributors are well-known transitive dependencies (`tar`, `semver`, `underscore`, `glob-parent`, `postcss`, `nth-check`, `webpack-dev-server`) inherited via `react-scripts` and `craco`.

Class	Count	Mitigation in our build
Build tooling	38	Build-time only; never runs in user browser. CSP blocks any escape.
Runtime dev server	7	`react-scripts start` ONLY runs in dev; production uses static `build/`.
Runtime (browser)	2	XSS-adjacent CVEs in old `nth-check` / `prismjs`; CSP `script-src` blocks all the relevant injection vectors.

Plan: scheduled craco/react-scripts bump in Q1 maintenance — most advisories disappear with that single bump.

Process

Monthly cadence:

1. `pip-audit -r backend/requirements.txt`
2. `yarn audit --json`
3. `cyclonedx-py requirements backend/requirements.txt --output-format JSON > sbom-backend.json`
4. Review CVE counts in this doc.
5. Bump any package with criticality ≥ "High" AND a fix available.
6. Update this doc with the new baseline.

Risk acceptance

All issues above are formally risk-accepted by the Security Lead until the next maintenance window (≤30 days from this baseline). Acceptance is documented in this file (NIST RA-3 compliant).

Reviewed: 2026-02-12 · Next: 2026-03-12 (monthly)

Data Processing Addendum (DPA) — Template

Data Processing Addendum (DPA) — Template

> **Status:** Template. This DPA forms part of the Master Services > Agreement (MSA) between Vigilant Entities ("Processor") and the > Customer ("Controller"). Defined terms not used here have the meaning > ascribed in the MSA. **Version 1.0 — effective on countersignature.**

1. Subject Matter & Duration

1.1 **Subject matter.** Processing of Customer Personal Data by Vigilant Entities for the sole purpose of providing the Amw■l OS / Vigilant Entities platform (the "Services"). 1.2 **Duration.** This DPA remains in effect for the duration of the underlying MSA plus any post-termination data return / deletion period required by Section 11.

2. Nature & Purpose of Processing

Item	Description
Nature	Hosting, storage, transmission, redaction, and analytical processing of Controller data through the Services.
Purpose	Provision of the Services as described in the Order Form / MSA.
Categories of Data Subjects	Controller's employees, contractors, customers, prospects, founders, investors, partners, and authorised end-users.
Categories of Personal Data	Identification (name, email, phone, role), authentication data (hashed), business records (deals, valuations, narratives), IP / device metadata, audit-log evidence.
Special Categories	None unless explicitly uploaded by Controller (Controller responsibility).

3. Controller & Processor Obligations

3.1 **Controller** warrants that it has a lawful basis for the Processing of Personal Data and that any instructions to Processor comply with Data Protection Laws. 3.2 **Processor** shall:

- Process Personal Data only on documented instructions from the

Controller (the MSA, this DPA, configuration in the Services).

- Ensure that personnel authorised to Process Personal Data are

bound by appropriate confidentiality obligations.

- Implement the technical and organisational measures set out in

Annex II ("Security Measures").

- Assist Controller in fulfilling its obligations to respond to Data

Subject requests (Articles 12–22 GDPR).

- Assist Controller with its obligations under Articles 32–36 GDPR

(security, breach notification, DPIA, prior consultation).

4. Sub-processors

4.1 Controller grants general written authorisation for Processor to engage Sub-processors listed in Annex III ("Sub-processor Register"). 4.2 Processor shall notify Controller of any intended addition or replacement of Sub-processors at least **thirty (30) days** in advance, by updating the Sub-processor Register at `https://vigilantentities.com/trust`. 4.3 Controller may object on reasonable grounds within fifteen (15) days; Processor shall use commercially reasonable efforts to accommodate the objection, failing which Controller may terminate the affected portion of the Services for cause without penalty.

5. Cross-Border Transfers

5.1 Personal Data is stored in **GCC region data centres** by default. EU / UK Personal Data, if any, is processed under the **2021 Standard Contractual Clauses** (Module 2 — Controller-to-Processor), which are incorporated by reference into this DPA. The UK IDTA addendum applies to UK transfers.

6. Data Subject Rights

6.1 Processor shall, to the extent legally permitted, promptly notify Controller if it receives a request from a Data Subject. 6.2 Processor shall provide self-service tooling in the Services (the "Privacy Center") allowing Controller to export, rectify, restrict, and erase Personal Data without Processor intervention.

7. Personal Data Breaches

7.1 Processor shall notify Controller without undue delay, and in any event within **forty-eight (48) hours**, of becoming aware of a Personal Data Breach. Notice shall describe (a) the nature of the breach, (b) likely consequences, (c) measures taken or proposed, (d) the point of contact. 7.2 Processor shall maintain a contemporaneous record of the breach, its containment, eradication, and recovery in the audit chain.

8. Audits & Inspections

8.1 Processor makes available all information necessary to demonstrate compliance with Article 28 GDPR, in the form of:

- the Security Overview, Threat Model, RBAC Matrix, Incident Response

Playbook, Backup & Restore Runbook, Vendor Risk Register, and Dependency Audit (all downloadable from `/trust`);

- CycloneDX SBOM for backend and frontend;
- Live audit-chain verifier at `/api/trust/verify`.

8.2 On reasonable notice and at Controller's expense, Processor will permit one (1) audit per twelve-month period by a mutually agreed independent auditor bound by confidentiality.

9. Confidentiality

9.1 Each party shall protect Personal Data and Confidential Information with at least the same degree of care it uses to protect its own Confidential Information, but in no event less than reasonable care.

10. Liability & Indemnification

Liability under this DPA is governed by the limitation of liability clause in the MSA, except where Data Protection Laws explicitly forbid limitation.

11. Return & Deletion

11.1 On termination of the MSA, Processor shall, at Controller's choice, return or securely delete all Personal Data within **thirty (30) days**, except where retention is required by law. Cryptographic audit-log entries necessary to evidence integrity may be retained for the period mandated by applicable record-keeping obligations.

12. Governing Law

This DPA is governed by the laws stipulated in the MSA. To the extent this DPA conflicts with the MSA in respect of Personal Data, this DPA prevails.

ANNEX I — Parties

Controller

- Legal name: _____
- Address: _____
- Authorised signatory: _____
- Title: _____
- Email (DPO / privacy lead): _____

Processor

- Legal name: Vigilant Entities (operating Amw■l OS)
- Authorised signatory: Syed Amer
- Title: Founder & Maker
- Email: security@vigilantentities.com
- DPA contact: privacy@vigilantentities.com

ANNEX II — Security Measures

Domain	Control
Encryption in transit	TLS 1.2+ (HSTS preload), ECDHE, modern ciphers
Encryption at rest	Storage-level AES-256 (managed DB), per-tenant logical isolation

Domain	Control
Access control	RBAC (system_admin / holding_admin / founder / investor / incubator_admin) enforced server-side
Authentication	bcrypt password hashes, magic-link sign-in, JWT with per-JTI revocation, brute-force lockouts
Audit logging	Hash-chained (`audit_chain.py`), publicly verifiable at `/api/trust/verify`
Device management	First-seen / last-seen / surgical per-device logout; suspicious sign-in alert emails
Network	Rate limiting, security headers (CSP, HSTS, X-Content-Type-Options, Referrer-Policy, Permissions-Policy)
Vulnerability mgmt	Quarterly dep-audit (pip-audit + yarn audit), CVE-driven SLA
Backups	Snapshot-based, encryption-at-rest preserved, tabletop drill log in `BACKUP_RUNBOOK.md`
Incident response	Documented playbook (`INCIDENT_RESPONSE.md`), 48-hour breach notification

ANNEX III — Sub-processors (as of issue date)

Vendor	Purpose	Data Categories	Region
MongoDB Atlas (or self-hosted)	Primary application database	All Customer Personal Data	GCC / configurable
Resend	Transactional email delivery (magic links, alerts, wrap-ups)	Email address, message metadata	EU / US
Stripe	Payment processing (subscriptions)	Billing contact, payment metadata (no card data)	US (PCI Level 1)
AWS (Bedrock + STS)	Internal AI inference (redacted)	Redacted, non-identifying tokens only	Configurable
OpenAI / Anthropic / Google (LLM inference)	LLM inference for redacted prompts	Redacted text only — never raw PII	US

> Always-current list: `https://vigilantentities.com/trust` → Vendor Risk > Register.

This template was generated from `/app/DPA_TEMPLATE.md`. Replace placeholder fields, countersign, and store the executed copy in your contract management system. For questions, contact **privacy@vigilantentities.com**.

Manifest & Integrity

Every document in this pack is reproducible. Recompute the SHA-256 of any source markdown file and compare to the row below. The bundle SHA-256 is the hash of the manifest itself.

Document	Size	SHA-256
Security Overview	9,415 B	`33fe37b0c953c3d552b276995cde38168959cb8cf05e6d9a090e5489f9b77a49`
Threat Model (STRIDE)	7,446 B	`290ad14166d531052cf83bddfe54f1dd2dd7df6e1ed559d586e5bf59f3811c75`
RBAC Matrix	6,401 B	`055e024cab63c498fbfa5848d10a9af94c874e7c98bdc889525458993ec133de`
Incident Response Playbook	5,780 B	`de9d6a4e0892e7a7e3149bbe03c9037a8b620f02a1ec5a889597beb2d832a6a1`
Backup & Restore Runbook	5,701 B	`9c0f103cf5ff4f94522376036f30c29a21bd3874663ef6e12818b6b1531a964e`
Vendor Risk Register	5,622 B	`3ad16405e3c12038745bf9de8ca7c8c9556bdc54d96616dbcd43ad3296ca798c`
Dependency Audit Baseline	3,874 B	`faedf0a1cca2efbc619f43429b7616aa53d59e22f625f2b3b44dcd9fc553d45c`
Data Processing Addendum (DPA) — Template	8,325 B	`dfca23bbc37c0e8262e3190eebb2fe673ceac86d9e8c693515fd4a3074961cf4`

Brand	`Vigilant Entities`
Generated	`2026-08-07 10:17 UTC`
Audit-chain head	`1c341c5613130ce8f789548faa916c4de2b5911d1ba42d567b7547e3878ef995`
Bundle SHA-256	`360142cc8a3ce744db162861a1402abfb7688dea46a5444b69b56a92542e9fa4`
PGP fingerprint	`2C38 50FA E800 5077 CC32 9B88 293B 5F6B 98E1 3975`
Public key	`/.well-known/security-pubkey.asc`
Live verifier	`/api/security/audit/verify/public`

This document is read-only evidence. For questions or to request a countersigned copy, email security@vigilantentities.com.

Appendix · Glossary

Plain-language explainers for the technical terms used throughout this pack. Designed for procurement and legal reviewers; technical readers may skip.

Bundle SHA-256

A cryptographic fingerprint of the entire evidence pack. We compute the SHA-256 hash of every source document, concatenate those hashes in a stable order, and hash the result again. If even one byte of any source changes — a typo fix, a date update — the Bundle SHA changes completely. Procurement teams should re-derive this value from their downloaded copy (we provide a one-click verifier on the Trust Center) and compare it to the value printed on the manifest page. Identical = the bundle has not been altered in transit.

SHA-256

A cryptographic hash function that converts any input — a file, a string, a binary blob — into a 64-character hexadecimal fingerprint. Two different inputs effectively never produce the same SHA-256 (collisions require computational effort that exceeds the lifetime of the universe). We use it to fingerprint every source document so any post-publication tampering is detectable.

Audit-chain head

The latest entry in our tamper-evident audit log. Each entry carries the hash of the previous entry, forming a linked chain — an attacker can't quietly rewrite history without breaking every downstream hash. The 'head' is the most recent entry's hash; publishing it pins this pack to a specific moment in our live system. A public verifier at /api/security/audit/verify/public walks the chain and reports any breakage.

PGP fingerprint

A short identifier (20 bytes, displayed as 10 pairs of hex characters) that uniquely names our cryptographic signing key. If you ever receive a copy of this pack countersigned with PGP, the signing key's fingerprint must match the one printed here. Mismatched fingerprint = either the wrong key was used or the pack has been forged. Our public key lives at /.well-known/security-pubkey.asc.

Tracking token

A unique short identifier (format: TKN-XXXXXXXXXXXX) embedded in every per-prospect copy of this pack. The token appears as a visible watermark on every non-cover page and in the PDF's metadata (Author, Subject, Keywords fields). If a copy of this pack later surfaces outside the intended recipient's organisation, we can trace it back to the original prospect via the token. The standard public pack has no token.

Defence-in-depth

Why we publish hashes, audit chains, PGP fingerprints, and per-prospect tokens all at once: each mechanism defeats a different attack. Hashes catch silent file-content tampering. The audit chain catches database manipulation. PGP signatures catch impersonation. Tracking tokens trace leaks. Together they make trustworthy supply-chain evidence — not a single chokepoint a sophisticated attacker could compromise to invalidate everything.