LiftedVFO Security Remediation Report

Date: {{DATE}} Prepared by: {{PREPARED_BY}}

Scope

This report addresses the findings and acceptance criteria outlined in "LiftedVFO - Security Pass 1.1 - 2025.09.18.docx". Each item includes remediation status, how it was addressed, and validation steps. Items not completed include next actions and any blockers.

Legend

- Status: Completed | In Progress | Needs Approval | Not Started | Needs Access

---

Priority 0

0.1 Replace browser tokens with secure cookie sessions

- Status: In Progress
- What we changed: Switched auth to HttpOnly cookie sessions; frontend uses withCredentials; removed Authorization header usage. Logged-in endpoints require cookie. Logout clears cookie.
- Notes/Follow-up: Confirm removal of any remaining token storage patterns; complete Google OAuth move fully server-side.
- Evidence: evidence/A1-browser-tokens.md
- Validation: {{VALIDATION_0_1}}
- Notes/Follow-up: {{NOTES_0_1}}

0.2 Encrypt provider refresh tokens at rest

- Status: In Progress
- What we changed: Added encrypted columns and admin backfill/drop endpoints. Awaiting APP_DEK and staging backfill run, then drop plaintext.
- Notes/Follow-up: Set APP_DEK (>=32 bytes) in Render; run backfill; verify; drop plaintext.
- Evidence: evidence/A2-token-encryption.md
- Validation: {{VALIDATION_0_2}}
- Notes/Follow-up: {{NOTES_0_2}}

0.3 Remove or sanitize raw HTML rendering

- Status: Completed
- What we changed: Replaced dangerouslySetInnerHTML with DOMPurify-powered SafeHtml in EditableSection/PublicBlog
- Validation: XSS payloads stripped in UI; unit tests to follow.
- Evidence: evidence/A3-xss-sanitization.md
- Validation: {{VALIDATION_0_3}}
- Notes/Follow-up: {{NOTES_0_3}}

0.4 Lock down uploads by size and type, stream safely

- Status: Completed
- What we changed: Enforced MIME allowlist and 20MB streaming guard in chat/legal endpoints
- Validation: >20MB -> 413; disallowed type -> 415; allowed under limit -> 200.
- Evidence: evidence/A4-upload-guards.md
- Validation: {{VALIDATION_0_4}}
- Notes/Follow-up: {{NOTES_0_4}}

0.5 CORS and secret key hardening

- Status: Completed
- What we changed: Restricted CORS, enumerated methods, and fail-fast SECRET_KEY/DATABASE_URL in prod
- Validation: Non-allowed origins blocked; app exits in prod if secrets missing/default.
- Evidence: evidence/A5-cors-secret.md
- Validation: {{VALIDATION_0_5}}
- Notes/Follow-up: {{NOTES_0_5}}

0.6 Replace admin header token with real RBAC

- Status: Completed
- What we changed: Removed X-Admin-Token, require session+admin role, added audit log + IP allowlist
- Validation: Admin routes 401/403 when not admin; audit logs present; optional IP allowlist enforced.
- Evidence: evidence/A6-admin-rbac.md
- Validation: {{VALIDATION_0_6}}
- Notes/Follow-up: {{NOTES_0_6}}

A7 Disable configuration probe endpoint in prod

- Status: Completed
- What we changed: Restricted /api/test/config to SuperAdmin only
- Validation: 403 for non-SuperAdmin; 200 for SuperAdmin.
- Evidence: evidence/A7-config-probe.md
- Validation: {{VALIDATION_A7}}
- Notes/Follow-up: {{NOTES_A7}}

A8 Remove hardcoded Google Client ID fallback

- Status: Completed
- What we changed: Removed literals, added prebuild env verification
- Validation: Frontend build fails without VITE_GOOGLE_CLIENT_ID.
- Evidence: evidence/A8-google-client-id.md
- Validation: {{VALIDATION_A8}}
- Notes/Follow-up: {{NOTES_A8}}

Priority 1

1.1 Brute force protection and admin 2FA

- Status: In Progress
- Changes: Added rate limiting on /api/token and TOTP enforcement when enabled; SuperAdmin enrollment endpoint; frontend OTP flow
- Notes: Complete admin enrollment UX; consider WebAuthn as next phase.
- Evidence: evidence/P1-1.1-bruteforce-2fa.md
- Validation: {{VALIDATION_1_1}}
- Notes: {{NOTES_1_1}}

1.2 RAG and AI safety guardrails

- Status: Not Started
- Changes: Planned: stricter prompt guardrails, tool allowlist enforcement, injection tests, source citation checks.
- Evidence: evidence/P1-1.2-rag-guardrails.md
- Validation: {{VALIDATION_1_2}}

- Notes: {{NOTES_1_2}}

1.3 Schema migrations via Alembic

- Status: Not Started
- Changes: Planned: Introduce Alembic versioned migrations and remove startup DDL.
- Evidence: evidence/P1-1.3-alembic.md
- Validation: {{VALIDATION_1_3}}
- Notes: {{NOTES_1_3}}

1.4 Security headers and CSP at the edge

- Status: Partial
- Changes: Added app-level security headers; edge CSP report-only ready with Render steps
- Notes: Apply CSP header in Render (report-only), monitor, then enforce.
- Evidence: evidence/P1-1.4-csp.md
- Validation: {{VALIDATION_1_4}}
- Notes: {{NOTES_1_4}}

1.5 Observability, health, and error reporting

- Status: Completed
- Changes: Added /healthz, DB-checked /readyz, request IDs, optional Sentry
- Validation: /healthz ok; /readyz returns ready; responses include X-Request-ID; Sentry captures errors when DSN set.
- Evidence: evidence/P1-1.5-observability.md
- Validation: {{VALIDATION_1_5}}
- Notes: {{NOTES_1_5}}

Priority 2

2.1 PII masking and selective encryption

- Status: In Progress
- Changes: Admin audit log redaction and global logging redaction filter; targeted encryption via 0.2.
- Notes: Extend masking coverage and lint checks; UI masking where appropriate.
- Evidence: evidence/P2-2.1-pii-masking.md
- Validation: {{VALIDATION_2_1}}
- Notes: {{NOTES_2_1}}

2.2 Dependency and image scanning in CI

- Status: In Progress
- Changes: Added Bandit, pip-audit, npm audit (prod), Trivy FS scan; fail on high/critical.
- Notes: Add container image scans and SBOM on releases.
- Evidence: evidence/P2-2.2-ci-scanning.md
- Validation: {{VALIDATION_2_2}}
- Notes: {{NOTES_2_2}}

2.3 Docker hardening

- Status: Completed
- Changes: Multi-stage builds; backend non-root user; nginx static serve for frontend.
- Evidence: evidence/P2-2.3-docker-hardening.md
- Validation: {{VALIDATION_2_3}}
- Notes: {{NOTES_2_3}}

Appendices

- Tracker: docs/security/Remediation-Tracker.csv
- Evidence pack (paths & excerpts): docs/security/evidence/
- Test plan: docs/security/Test-Plan.md

A1-browser-tokens.md

Evidence A1: Browser-stored tokens

Findings confirmed via grep:

```
frontend/src/apiClient.ts: localStorage.getItem('access_token') -> Authorization
header
frontend/src/pages/Login.tsx: localStorage.setItem('access_token')
frontend/src/services/googleCalendarService.ts:
localStorage.getItem('google_access_token')
```

Acceptance (target):

- No localStorage token usage
- HttpOnly cookie set on login
- withCredentials configured in Axios

A2-token-encryption.md

Evidence A2: Plaintext provider tokens at rest

Confirmed columns in `backend/app/models/user.py` :

```
google_access_token = Column(String, nullable=True)
google_refresh_token = Column(String, nullable=True)
```

Target remediation:

- Add `google_refresh_token_enc` and `google_refresh_token_iv`
- Backfill and null plaintext columns
- Drop plaintext columns after verification

Runbook:

1. Ensure `APP_DEK` (>=32 bytes) in Render backend env.
2. Login as Admin/SuperAdmin; run backfill:
   - POST `/api/admin/migrations/backfill-google-token-encryption`
   - Verify response `{ processed, skipped }`
3. Inspect DB: plaintext NULL; enc/iv populated.
4. Drop plaintext column:
   - POST `/api/admin/migrations/drop-plaintext-google-refresh-token`

A3-xss-sanitization.md

Evidence A3: Unsanitized HTML rendering

Occurrences:

```
frontend/src/components/modules/EditableSection.tsx: dangerouslySetInnerHTML
frontend/src/pages/PublicBlog.tsx: dangerouslySetInnerHTML
```

Target remediation:

- Introduce a `SafeHtml` component using DOMPurify
- Replace direct uses of `dangerouslySetInnerHTML`
- Add unit tests for `<script>` and `onerror` stripping

A4-upload-guards.md

Evidence A4: Upload endpoints lack size/type guardrails

Endpoints:

```
backend/app/api/chat.py: /upload-and-index
backend/app/api/legal.py: /entities/{entity_id}/documents/
```

Target remediation:

- ALLOW list: application/pdf, application/vnd.openxmlformats-officedocument.wordprocessingml.document
- MAX_BYTES = 20 * 1024 * 1024
- Stream reads with size guard; return 413/415 appropriately

A5-cors-secret.md

Evidence A5: CORS wide and default secret key present

Snippets:

```
backend/app/main.py: allow_origins includes multiple prod origins; allow_methods=["*"]
backend/app/core/config.py: SECRET_KEY = "your-secret-key-here"
```

Target remediation:

- Restrict prod `allow_origins` to official UI only
- Enumerate only required methods
- Read SECRET_KEY from env; fail fast in prod if missing

A6-admin-rbac.md

Evidence A6: Admin endpoints protected by shared header token

Snippets:

```
backend/app/api/admin_tasks.py: expects Header x_admin_token, compares to
ADMIN_TASKS_TOKEN
```

Target remediation:

- Remove header token auth
- Require authenticated session with admin role
- Add rate limiting and audit logging; optional IP allowlist via `ADMIN_IP_ALLOWLIST`

Additional controls implemented:

- Admin endpoints log structured "admin_action" entries with request ID and user info.
- Optional IP allowlist: set `ADMIN_IP_ALLOWLIST` to comma-separated IPs or CIDRs.

A7-config-probe.md

Evidence A7: Configuration probe endpoint exposed

```
backend/app/api/test_config.py: /api/test/config
```

Target remediation:

- Remove route in prod builds or restrict to SuperAdmin + internal network

A8-google-client-id.md

Evidence A8: Hardcoded Google OAuth client ID fallback

Occurrences:

```
frontend/src/providers/GoogleAuthProvider.tsx: import.meta.env.VITE_GOOGLE_CLIENT_ID
|| '<literal client id>'
frontend/src/pages/TestGoogleAuth.tsx: import.meta.env.VITE_GOOGLE_CLIENT_ID ||
'<literal client id>'
```

Target remediation:

- Remove literals; require env var at build time

P1-1.1-bruteforce-2fa.md

Evidence P1-1.1: Brute force rate limiting and 2FA

Implemented:

- Rate limiting on /api/token (10 attempts per 15 minutes per IP/username)
- TOTP 2FA enforcement for users with twofa_enabled set
- SuperAdmin endpoint to setup TOTP and return otpauth URI: POST /api/superadmin/2fa/setup

Planned next:

- Admin 2FA (TOTP/WebAuthn) enrollment and verification

P1-1.4-csp.md

Evidence P1-1.4: Security headers and CSP

Implemented at app layer:

- X-Content-Type-Options: nosniff
- Referrer-Policy: strict-origin-when-cross-origin
- X-Frame-Options: DENY
- Strict-Transport-Security: set when HTTPS

Recommended at edge (Render proxy / CDN):

- Content-Security-Policy (start in report-only) with script-src 'self' and allowed CDNs as needed.

Runbook:

1. In Render, open the frontend service -> Settings -> Headers -> Add Response Header:
   - Name: Content-Security-Policy-Report-Only
   - Value: default-src 'self'; script-src 'self' https://accounts.google.com https://apis.google.com https://cdn.lawpay.com https://secure.lawpay.com; style-src 'self' 'unsafe-inline' https://secure.lawpay.com; img-src 'self' data: https:; frame-ancestors 'none'; frame-src 'self' https://secure.lawpay.com https://www.youtube.com https://www.youtube-nocookie.com https://youtube.com https://drive.google.com https://docs.google.com; connect-src 'self' https://secure.lawpay.com https://cdn.lawpay.com https://agentiq-vfo-backend.onrender.com https://accounts.google.com https://apis.google.com; upgrade-insecure-requests
2. Use the app normally and watch Render logs/DevTools for CSP reports/violations.
3. Update the value to enforce CSP (change to Content-Security-Policy) once violations are addressed.

P1-1.5-observability.md

Evidence P1-1.5: Observability, health, and error reporting

Implemented:

- /healthz (static ok) and /readyz (DB ping)
- Request ID middleware adds X-Request-ID
- Optional Sentry integration via SENTRY_DSN

Runbook:

1. Set `SENTRY_DSN` in Render env to enable Sentry.
2. Confirm /readyz returns `{ "status": "ready" }` when DB reachable; `{ "status": "degraded" }` otherwise.
3. Confirm responses include X-Request-ID.

P2-2.1-pii-masking.md

Evidence P2-2.1: PII masking and selective encryption

Implemented:

- Admin audit logs redact sensitive fields (email, tokens, etc.).
- Masking strategy: emails partially masked; tokens show only prefix/suffix.

- Global logging redaction filter applied to root and uvicorn loggers.

Next:

- Extend masking to generic app logs (structured logging format with filters).
- UI masking for emails where appropriate.

P2-2.2-ci-scanning.md

Evidence P2-2.2: CI security scans

Added GitHub Actions workflow `.github/workflows/security.yml` to run:

- Bandit static analysis for backend
- pip-audit on backend dependencies
- npm audit for frontend (prod deps)
- Trivy filesystem scan, failing on HIGH/CRITICAL

Acceptance: CI fails the PR if high/critical findings are present.

P2-2.3-docker-hardening.md

Evidence P2-2.3: Docker hardening

Backend:

- Multi-stage build
- Non-root user `app`

Frontend:

- Multi-stage build (builder + nginx runtime)
- Serves static files via nginx

Next: Add Trivy container scans in CI and minimize base images further as needed.