PUBLIC SAMPLE — REDACTED · client identity removed; ▮▮ bars mark removed identifiers · methodology & findings intact

ARI

FULL-STACK SECURITY REVIEW

SECURITY REVIEW — APP & DNS LAYER

[redacted] · Track T5 — [redacted] · Port · [redacted] -Vault dApps + domains

PROJECT	TRACK / TIER	DATE
▮▮▮▮▮▮	**T5 · App track**	**24 May 2026**
CHAINS	**CODE VERSION**	**CLASSIFICATION**
n/a (application + DNS)	**locked at kickoff (pending)**	**PUBLIC SAMPLE**

Disclaimer. This report describes a time-boxed technical security review of the materials identified in the Scope of Review as they existed at the referenced commit/version during the review window. It is a point-in-time assessment: subsequent changes to code, configuration, infrastructure, or dependencies are not covered. No security review can identify all vulnerabilities. This report does not constitute a guarantee, warranty, or insurance of any kind, and it is not an endorsement of the project, its business model, or its tokens. It is not investment, legal, or financial advice. The review does not include formal verification, continuous monitoring, or exhaustive novel economic attack modeling unless expressly stated in scope. [redacted] may publish or share this report in full, unmodified form, including this disclaimer. Partial reproduction that omits findings, statuses, or this disclaimer is not authorized. Vari accepts no liability for decisions made by third parties on the basis of this report.

00 EXECUTIVE SUMMARY

Three dApp frontends plus the domains that front the suite. No committed secrets were found (keys read from env; only .env.example present) and hardcoded token addresses are canonical mainnet. Admin surfaces are genuinely protected — Port uses SIWE wallet auth on mutating routes; [redacted]-Vault uses signature auth with rate limiting, zod validation, HMAC-signed sessions and audit logging. The material issues are an unauthenticated [redacted] email endpoint usable for brand-spoofed phishing, a fail-open admin session-secret default, and the absence of CSP/security headers on the two Next.js apps. On the DNS side the domains are DNSSEC-signed on [redacted] but the DS record was never added at the .finance registrar, so the chain of trust is broken and no resolver validates; DMARC is monitor-only and DKIM is unpublished.

CRITICAL	HIGH	MEDIUM	LOW	INFO
0	2	3	4	1

Highest-priority items: **PROTO-T5-01** Unauthenticated [redacted] email endpoint (brand-spoofed phishing / quota abuse); **PROTO-T5-02** Admin session secret fails open to an in-repo default in production.

01 SCOPE OF REVIEW

Three dApp frontends ([redacted]-dapp — SvelteKit; port-dapp — Next.js; [redacted]-vault-dapp — Next.js + embedded Express) and the DNS/DNSSEC posture for [redacted], [redacted] and [redacted].

`TRACK :: T5 · TIER :: App track · CHAINS :: n/a (application + DNS) · CODE :: locked at kickoff (pending)`

Out of scope: third-party dependencies (base libraries, external protocols, bridge/oracle endpoints), off-chain keeper/relayer infrastructure, and key management, assumed correct unless expressly listed. Test, mock and deploy-script files were read for context but are not the subject of the findings.

02 METHODOLOGY

AI-assisted broad analysis over the full source followed by targeted human review of business logic, value-moving paths, roles/permissions and higher-risk areas; cross-layer reconciliation; and live-state verification of the deployed settings (see §05). The review is manual + AI-assisted + static; it does not include formal verification or runtime fuzzing unless stated. Severity reflects exploitability + impact, never effort: **Critical** (direct path to loss of funds / full compromise) · **High** (exploitable, realistic conditions, material impact) · **Medium** (limited impact or unusual preconditions) · **Low** (defense-in-depth / best practice) · **Info** (observation).

03 FINDINGS

ID	SEVERITY	LAYER	FINDING
PROTO-T5-01	HIGH	api	Unauthenticated [redacted] email endpoint (brand-spoofed phishing / quota abuse)
PROTO-T5-02	HIGH	api	Admin session secret fails open to an in-repo default in production
PROTO-T5-DNS	MEDIUM	config	DNSSEC chain of trust broken — DS record missing at the .finance registrar

PROTO-T5-03	MEDIUM	config	No CSP or security headers on the two Next.js apps
PROTO-T5-04	MEDIUM	frontend	Dynamic HTML-injection sinks on non-static data
PROTO-T5-05	LOW	api	Spoofable referer-based access control on the [redacted] GraphQL proxy
PROTO-T5-06	LOW	api	Unauthenticated cron GET endpoints on the SvelteKit app
PROTO-T5-DMARC	LOW	config	Email auth weak: DMARC p=none and DKIM not published
PROTO-T5-07	LOW	api	Test library (supertest) and full Express app in the production request path
PROTO-T5-08	INFO	config	security.txt absent; SIWE domain binding omitted; build-time type/lint checks disabled

HIGH PROTO-T5-01 **Unauthenticated [redacted] email endpoint (brand-spoofed phishing / quota abuse)**

LOCATION

`port-dapp/src/app/api/send-email/route.ts (POST)`

DESCRIPTION

The POST handler sends email via [redacted] from a [redacted]-verified sender with NO authentication — it only checks that template/to/values are present. The recipient is attacker-controlled and template values are interpolated unescaped into the HTML body. The SvelteKit equivalent requires a token header; this one does not.

IMPACT

Anyone can cause [redacted]'s verified sender to deliver “Deposit Confirmed” / “Withdrawal Request” emails to arbitrary recipients — high-credibility phishing under [redacted]'s brand/domain, plus [redacted] quota abuse and sender-reputation damage.

RECOMMENDATION

Require authentication (shared-secret header like the Svelte route, or session) before sending; restrict recipients to the authenticated user; rate-limit; HTML-escape interpolated values.

Status: OPEN · Confidence: high

HIGH PROTO-T5-02 **Admin session secret fails open to an in-repo default in production**

LOCATION

`[redacted]-vault-dapp/lib/auth/admin-session.js`

DESCRIPTION

If ADMIN_SESSION_SECRET is unset in production the code only logs an error and then derives the session secret from a hardcoded fallback string in the source tree — it does not fail closed. Admin session tokens are HMAC-signed with this secret.

IMPACT

If the env var is ever missing/misconfigured in prod, the HMAC key is public source, so anyone can forge a valid admin session token and pass admin auth — full admin API access (config, payments, withdrawals).

RECOMMENDATION

Fail closed: refuse to boot (or reject all admin auth) when the secret is unset in a prod runtime; enforce a minimum length; never fall back to a static default outside local dev.

Status: OPEN · Confidence: medium — exploitable only if the env var is missing; confirm it is set in prod

MEDIUM PROTO-T5-DNS

DNSSEC chain of trust broken — DS record missing at the .finance registrar

LOCATION

`[redacted] ([redacted] DNS) → .finance delegation`

DESCRIPTION

The zone is fully DNSSEC-signed (DNSKEY alg-13, valid RRSIGs on all records) but no DS record exists at the .finance parent, confirmed by authenticated NSEC denial from the TLD. Every validating resolver therefore returns AD=false and treats the domain as insecure — the signing effort currently buys nothing, and vest./vaults. inherit the same broken anchor.

IMPACT

DNS spoofing / cache-poisoning protections that a valid chain would provide do not apply; a hijack of DNS answers would not be detected by validators.

RECOMMENDATION

Export the DS from [redacted] (DNSSEC panel) and add it at the .finance registrar, then re-validate with DNSViz/delv. One DS secures the apex, vest. and vaults. together.

Status: OPEN · Confidence: high — verified via authenticated DoH

MEDIUM PROTO-T5-03

No CSP or security headers on the two Next.js apps

LOCATION

`port-dapp/next.config.ts ; [redacted]-vault-dapp/next.config.ts (no headers block)`

DESCRIPTION

Neither Next.js app defines a headers() config or [redacted].json headers — no CSP, X-Frame-Options/frame-ancestors, X-Content-Type-Options or Referrer-Policy. The SvelteKit app sets these; the Next apps (vaults. and vest.) do not.

IMPACT

No defense-in-depth against XSS, clickjacking or MIME sniffing on [redacted] and [redacted] — which raises the impact of the HTML-injection sinks below.

RECOMMENDATION

Add a headers() block (or [redacted].json headers) with a strict CSP, frame-ancestors none, nosniff, Referrer-Policy and HSTS; confirm against deployed responses.

Status: OPEN · Confidence: high — confirm on deployed sites

MEDIUM PROTO-T5-04 **Dynamic HTML-injection sinks on non-static data**

LOCATION

`[redacted]-dapp borrowers/[slug] {@html description}; [redacted] / Tooltip {@html prop}; port-dapp SideBySideText dangerouslySetInnerHTML`

DESCRIPTION

Several sinks render non-literal values as raw HTML (borrower profile description, banner/tooltip props, an icon prop). Most other {@html} uses are hardcoded strings; these carry data whose provenance is a backend/CMS or prop.

IMPACT

If any of these (especially borrower “description”, likely CMS/admin-sourced) can be influenced by a lower-trust actor or a compromised backend, this is stored XSS — and with no CSP there is no secondary mitigation.

RECOMMENDATION

Sanitize (e.g. DOMPurify) before injecting, or render as text; confirm the data source and who can set HTML for borrower descriptions.

Status: OPEN · Confidence: medium

LOW PROTO-T5-05 **Spoofable referer-based access control on the [redacted] GraphQL proxy**

LOCATION

`port-dapp/src/app/api/[redacted]/route.ts (validateRequest)`

DESCRIPTION

For non-server requests the proxy authorizes on the client-supplied Referer via substring matching, which is trivially spoofable and also matches lookalikes. The privileged master token is only attached to server requests, so this is limited to unauthenticated read-query relay/abuse.

IMPACT

Anyone can proxy arbitrary GraphQL read queries to the [redacted] backend (abuse/relay), not privilege escalation.

RECOMMENDATION

Do not rely on Referer; use exact-host Origin allow-listing or a signed token, and rate-limit.

Status: OPEN · Confidence: high

LOW PROTO-T5-06 **Unauthenticated cron GET endpoints on the SvelteKit app**

LOCATION

`[redacted]-dapp/src/routes/api/crons/coin-prices/+server.ts`

DESCRIPTION

The coin-prices cron GET performs no authorization before calling the paid CoinMarketCap API and writing to the DB; [redacted]-Vault protects its crons, this app does not.

IMPACT

Anyone can repeatedly trigger it to burn the paid API quota and force DB writes — resource/cost abuse (data itself is public).

RECOMMENDATION

Gate cron routes behind a CRON secret / Authorization check with constant-time comparison.

Status: OPEN · Confidence: high

LOW PROTO-T5-DMARC **Email auth weak: DMARC p=none and DKIM not published**

LOCATION

`_dmarc.[redacted] ; google._domainkey.[redacted]`

DESCRIPTION

DMARC is monitor-only (p=none) so spoofed mail is not quarantined/rejected (reporting to [redacted] is set). DKIM at the Google selector returns authenticated non-existence — [redacted] DKIM is not enabled, so outbound mail is unsigned and DMARC alignment is weak. SPF is ~all (softfail).

IMPACT

Weak anti-spoofing for the [redacted] domain — relevant given DeFi phishing and the unauthenticated email endpoint above.

RECOMMENDATION

Enable and publish [redacted] DKIM; after reviewing DMARC reports move to p=quarantine then p=reject; tighten SPF to -all once aligned.

Status: OPEN · Confidence: high — verified via DoH

LOW PROTO-T5-07 **Test library (supertest) and full Express app in the production request path**

LOCATION

`[redacted]-vault-dapp/app/api/[[...path]]/route.ts`

DESCRIPTION

The catch-all route imports the entire Express backend and dispatches every request through the supertest testing library, forwarding all client headers except host/content-length into the in-process app.

IMPACT

A dev/test dependency runs on every production API call (reliability/perf/maintenance risk) and blanket header forwarding widens the trusted-input surface.

RECOMMENDATION

Replace supertest with a real in-process handler or server; restrict forwarded headers; ensure the app does not trust client X-Forwarded-For for rate-limit/IP logic.

Status: OPEN · Confidence: medium

INFO PROTO-T5-08 **security.txt absent; SIWE domain binding omitted; build-time type/lint checks disabled**

LOCATION

`/.well-known/security.txt (404); port-dapp auth.ts (SIWE no domain); next.config ignoreBuildErrors / ignoreDuringBuilds`

DESCRIPTION

No security.txt is published for vulnerability disclosure. Port's SIWE verification omits domain binding (mitigated by nonce + admin-address + signature). Both Next apps disable ESLint and Port disables TypeScript build errors, so type/lint regressions can ship.

IMPACT

Weaker phishing resistance for admin SIWE and no published disclosure contact; loss of a compile-time safety net.

RECOMMENDATION

Publish a security.txt (Contact/Expires/Policy); bind the expected domain in SIWE; re-enable type/lint checks in CI.

Status: OPEN · Confidence: medium

04 WHAT'S SOLID

- No secrets committed: private keys / API keys / tokens all read from env; only .env.example templates in the repos.
- Port admin mutating routes all call SIWE authenticate() before proxying; [redacted]-Vault admin API enforces signature auth + rate limiting + zod schemas + timing-safe comparisons + audit logging.
- [redacted]-Vault cron endpoints protected by constant-time secret comparison; admin sessions HMAC-signed with revocation and TTL.
- Hardcoded token addresses verified canonical (USDC/USDT/DAI/WETH/wstETH mainnet).
- The SvelteKit app ships security headers via [redacted].json (CSP frame-ancestors none, nosniff, X-Frame-Options, Referrer-Policy) and gates its email route behind a token header.
- CAA is comprehensive across the intended CAs; authenticated-denial (NSEC) responses are consistent (no SERVFAIL anomaly).

06 SEAL OPERATIONAL SCORECARD

DIMENSION	STATE	DETAIL
Well-maintained framework	PASS	SvelteKit + Next.js, current; no abandoned core deps observed in source.
Dependency hygiene	VERIFY	Confirm lockfiles/audit in CI; supertest in the prod path (PROTO-T5-07).
Third-party scripts	VERIFY	Confirm on deployed pages (analytics/wallet SDKs) against CSP.
XSS sinks	ACTION	{@html}/dangerouslySetInnerHTML on non-static data — PROTO-T5-04.
Security headers	ACTION	Present on [redacted]; absent on vaults./vest. — PROTO-T5-03.
Content Security Policy	ACTION	No CSP on the two Next.js apps — PROTO-T5-03.
DNS & domain security	ACTION	DNSSEC chain broken (DS missing); DMARC p=none; DKIM unpublished — PROTO-T5-DNS/DMARC.
DDoS & availability	PASS	[redacted] DNS + [redacted] edge front the apps.
Backend access control	PASS/VERIFY	SIWE + signature auth on admin routes; one fail-open secret default (PROTO-T5-02) and one unauth email route (PROTO-T5-01).
Wallet interaction safety	VERIFY	Confirm tx-wrapper revert handling and chainId checks on the deployed flows.
Vulnerability disclosure	ACTION	No security.txt published — PROTO-T5-08.
Monitoring	VERIFY	Confirm DNSSEC/cert/uptime monitoring and alerting.
Account security	VERIFY	Hardware-key 2FA + registrar transfer-lock across registrar/[redacted]/[redacted]/[redacted] — highest-signal, must be confirmed.

07 DNSVIZ DNSSEC CHAIN-OF-TRUST

CHAIN LINK	STATUS	DETAIL
root → .finance	SECURE	TLD is signed; DS present in the root.
.finance → [redacted] (DS)	BROKEN	No DS at the registrar — authenticated NSEC denial from the TLD confirms none exists.
[redacted] apex keys	SIGNED but ISLANDED	DNSKEY 256+257 alg-13 (ECDSA-P256), valid RRSIGs on A/TXT/MX/NS/SOA — but unanchored, so AD=false.
Authenticated denial (NSEC)	HEALTHY	[redacted] NSEC responses consistent; no SERVFAIL/EDE anomaly.
vest. / vaults. ([redacted] CNAMEs)	INHERIT BROKEN ANCHOR	In-zone, signed by the same keys — one DS secures all three names.

Fix menu: export the DS from [redacted] and add it at the .finance registrar to complete the chain (one DS covers apex + vest. + vaults.); then re-validate with DNSViz / `delv`. Pair with a signed CAA, publish a security.txt, enable [redacted] DKIM, and move DMARC to enforcement. Add DS/DNSKEY + authenticated-denial monitoring so a failed rollover pages on-call before resolvers SERVFAIL.

08 ACTION POINTS

PRIORITY	ID	ACTION	STATUS
P0	PROTO-T5-01	Unauthenticated [redacted] email endpoint (brand-spoofed phishing / quota abuse)	◦ OPEN
P0	PROTO-T5-02	Admin session secret fails open to an in-repo default in production	◦ OPEN
P1	PROTO-T5-DNS	DNSSEC chain of trust broken — DS record missing at the .finance registrar	◦ OPEN
P1	PROTO-T5-03	No CSP or security headers on the two Next.js apps	◦ OPEN
P1	PROTO-T5-04	Dynamic HTML-injection sinks on non-static data	◦ OPEN
P2	PROTO-T5-05	Spoofable referer-based access control on the [redacted] GraphQL proxy	◦ OPEN
P2	PROTO-T5-06	Unauthenticated cron GET endpoints on the SvelteKit app	◦ OPEN
P2	PROTO-T5-DMARC	Email auth weak: DMARC p=none and DKIM not published	◦ OPEN
P2	PROTO-T5-07	Test library (supertest) and full Express app in the production request path	◦ OPEN
P2	PROTO-T5-08	security.txt absent; SIWE domain binding omitted; build-time type/lint checks disabled	◦ OPEN

P0 = before deploy / now · P1 = soon · P2 = hardening. Status column is filled at the re-review round (report goes to v1.1: ◦ OPEN · ✔ FIXED · ~ ACKNOWLEDGED · ✗ NOT FIXED).

09 RESIDUAL RISK

After the recommended fixes, residual risk concentrates in the privileged roles (owner/manager/strategist/governor keys) and in operational posture — key custody, multisig thresholds, timelocks, and monitoring — which code alone cannot guarantee. This is a point-in-time review; re-verify after any change to code, configuration, or deployed settings.