

TECHNICAL SECURITY REVIEW

Launch Pad Security Assessment

Smart contracts, migration, wallet frontend, points service, operations and npm supply chain

ASSESSMENT SUMMARY

11 FINDINGS IDENTIFIED

Technical observations and suggested remediation for the project team's review.

Repository

github.com/giupy997/launch-pad

Audited commit

27959e98d42e66cbddaf3a07b0317f8e3d628660

Assessment date

12 October 2026

1. Executive summary

Technical assessment for review by the project team.

Assessment context

No critical public, unauthenticated smart-contract drain was identified, and the contract test suite is unusually substantive. The review identified four High findings: the migration commitment is not enforced on-chain; freeze-mode fee harvesting creates an avoidable MEV window; the wallet application has known High npm advisories; and the production build path does not enforce the supplied npm security policy. This report provides technical findings and suggested remediation for the project team's evaluation. It does not represent or imply a launch, approval, endorsement or governance decision by LitVM or any other party.

4	4	3	185
High findings	Medium findings	Low findings	Foundry tests passed

What is working well

- Core contracts use reentrancy guards, checks-effects-interactions sequencing, SafeERC20, explicit fee and tax caps, and bounded transfer hooks.
- The cashback accounting uses conservative debt rounding and is covered by targeted solvency and exploit-regression fuzz tests.
- Pool pre-seeding and graduation edge cases receive dedicated defensive logic and tests.
- The web image proxy has meaningful SSRF controls: DNS resolution, private-range rejection, redirect denial, content and byte limits, and edge rate limiting.
- The current lock resolves only to the public npm registry, includes integrity data for direct packages, and passed npm registry signature verification.

Priority review areas

Area	Suggested remediation outcome	Suggested evidence
Migration integrity	Every holder allocation and coin total is cryptographically bound to the committed snapshot root.	Proof tests plus an independently reproduced snapshot.
Economic execution	Freeze and harvest cannot be permissionlessly forced through an unbounded, sandwichable sale.	Fork test and adversarial MEV simulation.
Dependencies	High advisories are removed or narrowly documented as non-applicable; direct versions are exact.	Dedicated dependency PR, audit JSON and build output.
Deployment governance	The production pad is owned by the intended 48-hour timelock without an EOA custody gap.	Verified addresses, roles, bytecode and ceremony transcript.

2. Scope, snapshot and methodology

A code-and-configuration assessment, not an attestation of a future deployment.

Area	Included	Depth
Solidity	Launchpad, LaunchpadBase, LaunchpadMigration, LaunchToken, factory, migrators and zap router	Manual review, tests, build sizes, threat modeling
Migration and operations	Freeze, snapshot, replay, timelock and deployment scripts	Privilege and invariant analysis
Wallet web app	Next.js routes, swap flow, RPC/image proxy, headers and client configuration	Manual review, unit tests, lint and production build
Points service	Indexer, append-only store, referrals, API and systemd deployment	State-machine and operational review
Supply chain	web/package.json, package-lock.json, npm metadata, lifecycle scripts and repository controls	Immutable install with scripts disabled, audit and signature checks
Live observation	Documented Liteforge rehearsal pad only	Read-only RPC calls; no transaction was sent

Snapshot

The audited repository was cloned from <https://github.com/giupy997/launch-pad> at commit 27959e98d42e66cbddaf3a07b0317f8e3d628660, authored 11 October 2026. Git submodules were checked out at their committed immutable SHAs. The review did not alter tracked repository files.

Method

- Manual data-flow, authorization, accounting, state-transition and external-call review.
- Adversarial analysis of migration, freeze, graduation, AMM execution, wallet transactions, RPC trust, persistence and CI/build behavior.
- Foundry compilation, 185 contract tests and size checks.
- A sanitized-environment npm ci --ignore-scripts, 27 Node tests, lint, production build, npm audit and npm audit signatures.
- Registry metadata scan for the seven-day release hold across exact lockfile versions.

Limitations

- No formal verification, independent differential specification, private-key ceremony review, or production LitVM bytecode existed in scope.
- Dormant Litecoin/Zcash experiments were assessed for repository exposure but were not line-by-line audited because the README states they are not built, run or deployed.
- Third-party protocols, wallets, RPC providers, the Lester Labs router and future LitVM consensus/bridge behavior were not audited.
- A passing test suite and clean signature result reduce risk but do not prove the absence of malicious or vulnerable dependency code.

Live rehearsal observation

At LitVM Liteforge chain 4441, block 60,254,114, the documented v11 rehearsal pad 0x39D104b3258B6A18c5d5d967CDA182Ded20Bef7F had 23,615 bytes of code, feeBps 100, tokenCount 5, freezeBlock 0 and migrator 0xD45e...13F. Its owner 0x707f...9A02 had no bytecode and was therefore an EOA at the observation point. This validates the README's rehearsal description; it is not the proposed mainnet deployment.

3. Findings register

Severity reflects impact and practical exploitability in the stated LitVM production context.

ID	Severity	Finding	Status	Area
H-01	High	Migration root is informational, not enforced	Open	Migration / contracts
H-02	High	Freeze-mode harvest exposes unbounded public AMM sale	Open	Contracts / MEV
H-03	High	Known High npm vulnerabilities remain in wallet build	Open	Web / dependencies
H-04	High	Production npm path lacks mandatory supply-chain controls	Open	Build / operations
M-01	Medium	LitVM deployment has a non-atomic EOA governance handoff	Open	Deployment
M-02	Medium	Points checkpoint crash can duplicate credited events	Open	Points service
M-03	Medium	Two-step swap can spend wallet ETH beyond sale proceeds	Open	Wallet UX / funds
M-04	Medium	Repository lacks CI security and protected-file controls	Open	Governance
L-01	Low	Quote-asset accounting assumes exact ERC-20 transfers	Open	Contracts
L-02	Low	Systemd services omit practical sandbox directives	Open	Operations
L-03	Low	Browser policy retains unsafe-inline script allowance	Open	Web defense

Severity model

- High: plausible loss, misallocation, or build compromise with material user or protocol impact; prioritize for team review and remediation.
- Medium: material weakness requiring additional conditions, timing or user state; assess, remediate or formally accept with documented rationale.
- Low: defense-in-depth or operational weakness with constrained direct impact; schedule and track.

No Critical finding

The assessment did not identify a public, permissionless path to drain all pad reserves or mint arbitrary supply in the audited code. This is not a guarantee: the High migration and supply-chain findings can still cause severe user loss if left unresolved.

4. High findings

Priority technical findings for review, remediation and re-test.

H-01 / HIGH

Migration root is informational, not enforced

Impact. A designated migration operator can choose holder addresses and balances without presenting a proof that those values belong to the committed source snapshot. The module prevents duplicate holders and total over-allocation, but it does not prevent arbitrary reallocation within the aggregate. A compromised or mistaken operator could confiscate source-holder claims during Base-to-LitVM migration.

Evidence.

- `contracts/src/LaunchpadMigration.sol:43-47` stores `migrationRoot` and freeze height.
- `LaunchpadMigration.sol:73-113` accepts coin metadata, holders and balances from the authorized operator with no Merkle proof or committed calldata hash.
- `LaunchpadMigration.sol:182-200` enforces no duplicate holder and no aggregate overpayment, but cannot establish who owned each source balance.
- The README describes independent replay/compare checks; those checks are off-chain monitoring rather than destination-chain enforcement.

Recommendation. Commit to a canonical snapshot tree whose leaf domain includes source chain, source pad, source token, destination chain, holder and amount. Require per-holder Merkle proofs and a replay bitmap, and separately bind coin totals and metadata. If batch execution is required, commit the entire canonical migration file or calldata hash before execution and require independent signed attestations. Keep the operator unable to substitute recipients.

Closure criteria. Adversarial tests show that a valid aggregate with one altered holder or amount reverts; the published snapshot tool reproduces the on-chain root; an independent party verifies the final holder set and totals before unpausing user activity.

H-02 / HIGH

Freeze-mode harvest exposes an unbounded public AMM sale

Impact. As soon as a freeze is announced, any caller may harvest the entire accumulated tax buckets in one transaction, with the normal one-block cooldown and harvest cap removed. Because execution accepts the current pool quote without a caller-supplied minimum output, an attacker can sandwich the predictable sale and reduce quote assets distributed to creator, platform and cashback recipients.

Evidence.

- `contracts/src/UniV2Migrator.sol:303-315` documents the freeze rush path.
- `UniV2Migrator.sol:321-337` sets rush when `freezeBlock` is merely non-zero, removes the cooldown/cap and sells the full bucket.
- The callable path is public and lacks a minimum quote output or an oracle/TWAP deviation bound.
- The runbook recommends emptying buckets before announcement, but the contract does not enforce that invariant and taxed trades can accrue during the announcement window.

Recommendation. Keep harvest bounded until the freeze actually takes effect. Add a minimum-output or conservative TWAP/deviation check, and route migration-day maintenance through a controlled keeper/private transaction path. Alternatively require buckets below a hard threshold before `announceFreeze` and disable new accrual immediately. Model a hostile block builder in fork tests.

Closure criteria. A fork test demonstrates bounded loss under a sandwich attempt at maximum permitted bucket size; `announceFreeze` cannot create an unbounded public sale; operational documentation names the exact keeper and failure mode.

4. High findings (continued)

H-03 / HIGH**Known High npm vulnerabilities remain in the wallet build**

Impact. The audited lockfile reports 39 known vulnerabilities: 13 High and 26 Moderate. This includes a direct Next.js version with a High server-side request forgery advisory and transitive wallet/build packages with denial-of-service, prototype-pollution and request-routing advisories. The direct High advisory warrants priority remediation before production deployment.

Evidence.

- next@16.3.7: High image-optimization SSRF (GHSA-cjq9-62q9-8jv4); a fixed 16.3.8 release was published 30 September 2026 and is older than seven days.
- axios@1.16.0 is transitive through wagmi -> @wagmi/connectors -> @base-org/account -> @coinbase/cdp-sdk; audit reports several High advisories. The candidate fixed line includes 1.20.0, published 26 August 2026.
- ws@8.18.0 is transitive through WalletConnect/Reown nested viem and is affected by GHSA-96hv-2xvq-fx4p; 8.21.0 is outside the release hold.
- source-map-js@1.2.1 and nanoid@3.3.16 carry High denial-of-service advisories in build/runtime dependency paths.
- No critical advisory was reported, but audit absence is not evidence of safety and reachability was not formally proved for every transitive path.

Recommendation. Create a dedicated dependency-only pull request. Update Next.js first, then reduce or update the wagmi connector graph so axios/ws advisories disappear. Pin every direct dependency exactly, inspect lockfile and lifecycle-script deltas, run the full contract/web test suite and build, then rerun npm audit and signatures. Do not use npm audit fix --force and do not suppress advisories without path, reachability, controls, approver and expiry documentation.

Closure criteria. The release lock contains no applicable High or Critical advisory, or each exception is narrowly documented and approved. All selected releases satisfy the seven-day hold. Tests, production build, audit and signature verification pass in an isolated environment.

4. High findings / supply chain

H-04 / HIGH**Production npm path lacks mandatory supply-chain controls**

Impact. The repository does not enforce release age, lifecycle-script denial, exact direct versions or a pinned package-manager. The points deployment notes explicitly instruct npm install --no-audit --no-fund, and Netlify declares only npm run build. A clean deployment can therefore resolve new dependency versions and may execute dependency-controlled native/install hooks in the build environment.

Evidence.

- No project-level .npmrc exists. Required controls such as min-release-age=7, ignore-scripts=true, save-exact=true and strict-peer-deps=true are absent.
- Every direct dependency and devDependency in web/package.json uses a caret or other loose range; no exact packageManager field is present.
- points/deploy/notus-points.service comments recommend npm install --no-audit --no-fund rather than immutable npm ci --ignore-scripts.
- Five exact lock packages advertise lifecycle scripts: bufferutil@4.1.0, fsevents@2.3.3, keccak@3.0.4, unrs-resolver@1.12.2 and utf-8-validate@5.0.10.
- The dependency graph contains 1,092 packages and several deprecated wallet packages, magnifying review and compromise surface.

Recommendation. Commit the mandated .npmrc, exact direct versions and an approved exact npm packageManager version. Make CI, Netlify and deployment use npm ci --ignore-scripts; ensure the lock cannot be rewritten. Remove unused wallet connectors/SDKs to shrink transitive exposure. If any package truly requires an install hook, stop, inspect its exact tarball/script/provenance, isolate execution and obtain explicit narrow approval rather than weakening ignore-scripts.

Closure criteria. A clean isolated build succeeds with immutable installation and scripts denied; package.json and lockfile agree; no unreviewed lifecycle hook runs; deployment documentation and platform configuration use the same controlled command.

5. Medium findings

M-01 / MEDIUM

LitVM deployment has a non-atomic EOA governance handoff

Impact. The LitVM deployment script creates a pad owned by the deploying account, while a separate script later deploys the timelock and transfers ownership. During that gap, one EOA can immediately change fee, treasury, quote asset and migrator settings, appoint the migration operator, freeze the system or migrate it out. A compromised ceremony key can materially change the launch before governance controls take effect.

Evidence.

- `contracts/script/DeployLitVM.s.sol:57-68` deploys the pad with deployer ownership and stops.
- `DeployTimeLock.s.sol` performs the ownership transition as a later operation; README expects a 48-hour production delay.
- Read-only RPC inspection confirmed the documented v11 rehearsal pad remains owned by an EOA. This observation does not establish the intended production governance posture.

Recommendation. Deploy the production pad under the final `TimelockController` atomically, matching the safer Base deployment pattern. Use a multi-signature proposer, open executor only if intentional, no orphan admin, and a 48-hour minimum delay from the first funded block. Publish and independently verify roles, delay, owner, bytecode and constructor arguments.

Closure criteria. A single ceremony transaction or deterministically ordered batch leaves no funded state under EOA ownership; an automated script asserts every final role and owner before the frontend enables mainnet.

M-02 / MEDIUM

Points checkpoint crash can duplicate credited events

Impact. The points indexer appends processed events before persisting its block checkpoint. A process crash between those writes leaves durable events but an old checkpoint. On restart the same chain logs are replayed and appended again because the store has no unique transaction/log-index key. If points acquire economic or reputational value, users can receive duplicate credit without an on-chain duplicate.

Evidence.

- `web/lib/points/indexer.ts:152-155` appends events and then writes metadata/checkpoint state.
- `web/lib/points/store.ts:62-64` provides append behavior without durable event identity deduplication.
- The failure requires a narrow crash window but normal restarts make recovery deterministic and repeatable.

Recommendation. Make event application idempotent using `chainId`, `transactionHash` and `logIndex` as a unique key. Persist events and checkpoint atomically, or rebuild from an immutable deduplicated journal. Add a fault-injection test that crashes after append and before checkpoint, then verifies an identical score after restart.

Closure criteria. Crash-recovery and reorg tests demonstrate exactly-once effective credit, even when the raw log is replayed multiple times.

5. Medium findings (continued)

M-03 / MEDIUM**Two-step token swap can spend wallet ETH beyond sale proceeds**

Impact. The UI estimates the native asset expected from the sell transaction, then automatically prepares the buy leg using that estimate. If the sell clears at a lower but acceptable amount, the second transaction may consume pre-existing wallet native balance to cover the difference. If the wallet lacks the buffer, the flow stops after the sale and leaves the user in the native asset.

Evidence.

- `web/app/swap/page.tsx:118-159` derives `ethIn` from the quote and uses it as the second transaction value rather than measuring the confirmed balance delta.
- Lint also reports a missing React hook dependency for `minOut` at line 159, increasing stale-state risk in the same flow.

Recommendation. Prefer an atomic on-chain router with user-defined end-to-end minimum output. Otherwise measure the confirmed native balance delta, cap the buy to actual proceeds, re-quote after confirmation and require explicit confirmation if any wallet top-up would be spent. Fix the hook dependency and add a slippage regression test.

Closure criteria. A test where the sale realizes the minimum accepted output cannot spend unrelated wallet funds; partial completion is clearly presented and recoverable.

5. Medium findings (continued)

M-04 / MEDIUM

Repository lacks CI security and protected-file controls

Impact. There are no GitHub Actions workflows, dependency-review gate, automated audit/signature checks, secret scanning configuration, static analysis, security policy or CODEOWNERS. A contributor can change security-critical dependency/build/deployment files without repository-local evidence that the expected checks or specialist review ran.

Evidence.

- No `.github/workflows`, `SECURITY.md` or `CODEOWNERS` was present at the audited commit.
- Because there are no actions, no mutable action tags were found; when added, every third-party action must be pinned to a full commit SHA with a release comment.
- `forge fmt --check` currently fails, so a future quality gate needs a formatting cleanup or an explicitly scoped exception.

Recommendation. Add least-privilege CI with permissions: contents: read; immutable npm installation with scripts denied; tests, build, npm audit, npm audit signatures, Foundry tests and formatting; dependency review; static analysis and secret scanning. Protect `package.json`, `lockfiles`, `.npmrc`, `workflows`, `Docker/deploy` files and contracts with `CODEOWNERS` and mandatory review. Keep dependency updates in dedicated PRs.

Closure criteria. A clean pull request demonstrates all pinned, least-privilege checks; branch protection requires them; security-critical paths require named review; fork PRs receive no secrets.

L-01 / LOW

Quote-asset accounting assumes exact ERC-20 transfers

Impact. The buy path credits the requested amount rather than a measured balance delta. If governance whitelists a fee-on-transfer, rebasing or callback-heavy quote token, the curve can credit more value than received and become under-collateralized. The intended LitVM native asset and current Base token may not exhibit this behavior, making this a governance-triggered compatibility risk.

Evidence.

- `contracts/src/Launchpad.sol:162` uses the requested ERC-20 amount in accounting after `transferFrom`.
- `setQuoteAsset` does not enforce exact-transfer semantics; token behavior is delegated to governance due diligence.

Recommendation. Restrict quote assets to an explicit reviewed allow-list and document required token semantics. Prefer measuring the balance delta and reverting unless received equals requested. Do not support fee-on-transfer/rebasing assets without a separately specified accounting model.

Closure criteria. A mock fee-on-transfer quote token cannot create under-collateralized curve credit; governance documentation names the compatibility checks for each quote asset.

5. Low findings (continued)

L-02 / LOW

Systemd services omit practical sandbox directives

Impact. The supplied points/watcher unit files do not use common operating-system containment settings. A web or dependency compromise would inherit broader filesystem and process visibility than the service needs.

Evidence.

- NoNewPrivileges, ProtectSystem, ProtectHome, PrivateTmp, restrictive UMask and explicit ReadWritePaths are absent from the supplied service definition.
- The points service correctly binds to 127.0.0.1 behind Caddy, which limits direct network exposure.

Recommendation. Run as a dedicated unprivileged account and add systemd sandboxing compatible with the state directory. Keep secrets out of unit files, restrict writable paths and egress where practical, and verify restart behavior under the hardened profile.

Closure criteria. systemd-analyze security and integration tests show a materially restricted unit without breaking state persistence or health checks.

L-03 / LOW

Browser policy retains unsafe-inline script allowance

Impact. The application sends several useful browser headers, but its Content Security Policy allows inline scripts. This weakens CSP as a containment layer if a future rendering or dependency issue introduces script injection.

Evidence.

- The production header set includes HSTS, frame denial, MIME sniffing denial and permissions restrictions.
- The current CSP contains `script-src 'unsafe-inline'`; no dangerous HTML sink was found in the reviewed active application.

Recommendation. Adopt framework-supported nonces or hashes and remove unsafe-inline where deployment permits. Add a CSP reporting endpoint during rollout and keep connect-src restricted to explicitly required RPC/wallet origins.

Closure criteria. Production responses enforce a nonce/hash-based policy without console violations on all supported routes and wallet flows.

6. Suggested remediation roadmap

A staged sequence for the project team's evaluation and implementation.

Stage	Suggested work	Evidence to retain	Review checkpoint
1. Code fixes	H-01 proof enforcement; H-02 bounded/slippage-protected harvest; M-02 idempotent points; M-03 swap proceeds cap.	PRs, threat model, unit/fuzz/fork tests	No open High contract finding
2. Dependency PR	Exact versions; update Next and vulnerable paths; trim connectors; .npmrc; exact npm packageManager.	Release ages, tarball/script review where needed, lock diff, audit/signature JSON	No unresolved applicable High
3. CI controls	Pinned action SHAs, least privilege, immutable installs, audits, tests, dependency review, CODEOWNERS.	Passing protected checks and branch rules	No bypass for launch tag
4. Rehearsal	Fresh pad; reproduce snapshot root independently; simulate MEV and crash/reorg recovery.	Block numbers, roots, traces, independent sign-off	All acceptance criteria met
5. Ceremony	Atomic timelock ownership; verified roles/bytecode; multisig proposer; 48-hour delay; production RPC validation.	Signed checklist, Safe transaction links, explorer verification	Frontend remains disabled until verified
6. Monitoring	Alert on root/freeze/owner/operator/config changes, abnormal harvest slippage and indexer duplicates.	Runbook, paging test, rollback/pause authority	Owner accepts residual risk

Suggested production verification assertions

- The production chain ID, RPC, explorer, quote asset, router, treasury, fee, tax, graduation target and virtual reserve are reviewed as one signed manifest.
- The deployed runtime bytecodes match reproducible compiler output and constructor arguments; verified source is available before deposits.
- The pad owner is the intended timelock, proposer is the intended multisig, delay is at least 48 hours, and no unexpected admin remains.
- The migration root is independently reproduced and cryptographically enforced; total token supply and every claimant allocation reconcile.
- Harvest/AMM operations have explicit price protection and a tested failure path; no public call can force an unbounded sale.
- The wallet app points only to reviewed production contracts and does not silently fall back to rehearsal addresses.

Project team review note

This assessment does not make a launch, approval or governance decision on behalf of LitVM or any other party. The findings, severity ratings and remediation suggestions are provided for the project team to assess and action as it considers appropriate. A follow-up review of material remediation changes would provide additional assurance.

7. Verification and test evidence

Check	Result	Evidence
Foundry tests	PASS	185 passed, 0 failed, 0 skipped across 13 suites; includes fuzz/regression coverage.
Foundry build and sizes	PASS	Solc 0.8.24. Launchpad runtime 20,835 bytes, 3,741 bytes below EIP-170. Payable fallback/no receive warning noted.
Foundry formatting	FAIL	forge fmt --check reports diffs across source, tests and scripts. Quality issue; not treated as direct vulnerability.
Web tests	PASS	27 passed, 0 failed in the active web application.
Web lint	WARN	No errors; one missing hook dependency at web/app/swap/page.tsx:159.
Next production build	PASS/WARN	20 routes built. Bundler warned about a critical dynamic expression in ox tempo virtualMasterPool imported through viem.
npm immutable install	PASS	npm ci --ignore-scripts in a sanitized environment; 1,036 packages added; no lifecycle script executed.
npm audit	FAIL	39 total: 13 High, 26 Moderate, 0 Critical.
npm audit signatures	PASS	0 invalid and 0 missing registry signatures.
Release-age scan	PASS/WARN	838 of 839 unique exact lock versions dated; 0 published in the last seven days; one registry date unavailable.
Repository secrets scan	PASS	No tracked private key or live secret identified; only an environment example was tracked.
Live Liteforge read	INFO	Rehearsal pad exists and matches documented EOA ownership; no transaction sent.

Commands were intentionally constrained

Dependency installation ran with a clean environment and `--ignore-scripts`. No `npx`, `npm exec`, `pnpm dlx`, `audit fix`, `remote install script` or `dependency lifecycle hook` was executed. The audit did not send blockchain transactions or use credentials.

Compiler and test observations

- Contract runtime sizes remain below the EIP-170 ceiling at the audited compiler settings.
- A payable fallback without `receive()` warning is intentional in context: bare native transfers revert through module dispatch unless a supported selector is used.
- The production build warning originates in a transitive `viem/ox` module; it should be rechecked after connector pruning and dependency updates.

8. npm dependency and release-hold review

Existing direct versions resolved by the audited lockfile. No dependency was added by this assessment.

Package	Exact resolved	Published	Class
@netlify/blobs	11.1.1	2026-09-24	prod
@noble/curves	1.9.1	2025-05-14	prod
@noble/hashes	1.8.0	2025-04-21	prod
@scure/base	2.4.0	2026-08-28	prod
@scure/btc-signer	2.4.1	2026-08-29	prod
@tanstack/query-core	5.101.4	2026-07-21	prod
@tanstack/react-query	5.101.4	2026-07-21	prod
next	16.3.7	2026-09-29	prod / vulnerable
qrcode	1.5.3	2023-04-22	prod
react	19.3.0	2026-09-09	prod
react-dom	19.3.0	2026-09-09	prod
@types/react	19.3.0	2026-09-09	prod
@types/react-dom	19.3.0	2026-09-09	prod
server-only	0.0.1	2022-09-03	prod
viem	2.55.10	2026-07-27	prod
wagmi	2.19.5	2025-11-19	prod
@types/node	20.19.43	2026-06-10	dev
@types/qrcode	1.5.6	2025-10-24	dev
eslint	9.39.5	2026-07-10	dev
eslint-config-next	16.3.7	2026-09-29	dev / vulnerable path
postcss	8.5.23	2026-07-24	dev
tailwindcss	3.4.19	2025-12-10	dev / vulnerable path
typescript	5.9.3	2025-09-30	dev

Release-hold conclusion

Every dated exact version in the lockfile was older than seven days on 12 October 2026. The scan resolved publication dates for 838 of 839 unique exact package versions and found zero younger releases. One version could not be dated from registry metadata and remains a manual verification item. No release-age exception is required for the proposed Next.js fix or the representative fixed axios/ws versions because those releases are also older than seven days.

Important distinction

The lock currently resolves exact versions, but package . json still expresses loose direct ranges and the repository lacks a project-level min-release-age rule. The observed age result is therefore a snapshot fact, not an enforced future control.

8. npm dependency review (continued)

Representative High advisory paths

Package	Installed	Path	Advisory	Candidate	Risk
next	16.3.7	Direct	GHSA-cjq9-62q9-8jv4	16.3.8	Image optimization SSRF
axios	1.16.0	wagmi -> connectors -> base account -> CDP SDK	Multiple High	1.20.0	Prototype pollution / SSRF / DoS
ws	8.18.0	WalletConnect/Reown nested viem	GHSA-96hv-2xvq-fx4p	8.21.0	Memory exhaustion
source-map-js	1.2.1	postcss / Next / Tailwind	GHSA-68fv-2mzg-jv7q	1.2.2	Event-loop denial of service
nanoid	3.3.16	postcss	GHSA-2v37-7h3g-55p8	3.3.18	Infinite loop in custom generator
braces/micromatch	various	eslint-config-next / tailwind dev graph	High paths	Review majors	Pattern-processing denial of service

Lifecycle scripts detected

Package	Version	Lifecycle entry	Context
bufferutil	4.1.0	install: node-gyp-build	Native optional acceleration; transitive wallet graph
fsevents	2.3.3	package marked hasInstallScript; build/prepublish metadata	Platform watcher dependency
keccak	3.0.4	install: node-gyp-build exit 0	Native hashing implementation
unrs-resolver	1.12.2	postinstall: node postinstall.js	Resolver native package selection
utf-8-validate	5.0.10	install: node-gyp-build	Native optional validation

None of these scripts was executed during this assessment. Their presence is not proof of malicious behavior, but each would require exact-version review and explicit approval before any intentional execution.

Dependency graph observations

- The lockfile contains 1,092 packages: 564 production, 347 development and 210 optional entries as reported by npm audit metadata; categories overlap npm's internal accounting.
- Only walletConnect is imported from wagmi/connectors in the reviewed application, yet the graph includes Coinbase CDP, MetaMask SDK, Reown/AppKit, Safe SDK and optional native modules. Connector pruning is the highest-value dependency-reduction exercise.
- The immutable install reported a React 19.3 peer conflict against use-sync-external-store@1.2.0 expecting React 18 or earlier. The build passed, but the mismatch should be removed rather than normalized as warning noise.

9. Final supply-chain security report

Required close-out against the supplied Secure JavaScript and npm Software-Build Policy.

Control	Assessment result
Dependencies changed	No third-party dependencies were added, removed, or updated.
Exact versions involved	No version change was made. Current direct resolved versions are listed in section 8; flagged installed and candidate fixed versions are listed in the advisory table.
Why new dependencies were required	Not applicable; no new dependency was introduced.
Seven-day hold	0 of 838 dated exact lock versions were younger than seven days; 1 of 839 could not be dated and needs manual verification. No exception was used.
Lifecycle scripts	Five exact packages advertise lifecycle hooks: bufferutil@4.1.0, fsevents@2.3.3, keccak@3.0.4, unrs-resolver@1.12.2 and utf-8-validate@5.0.10.
Scripts executed	None. Installation used npm ci --ignore-scripts in a sanitized environment.
Lockfile changes	None. The existing package-lock.json was neither deleted nor rewritten for delivery.
Vulnerability audit	Audit result: 13 High, 26 Moderate, 0 Critical. No audit fix or automatic major update was run.
Signatures / provenance	npm audit signatures reported 0 invalid and 0 missing. Lock integrity and official registry URLs were present. Separate package provenance attestations were not established for every package.
CI / GitHub Actions	No workflows existed and none was changed. Therefore no mutable Action reference was introduced; required CI and dependency-review controls remain absent.
Remaining manual review	Confirm the one undated lock version, assess exact tarballs/scripts for any retained lifecycle package, validate advisory reachability, and independently review the production deployment ceremony.
Policy relaxation	No security rule was silently relaxed. The audit stopped short of executing scripts or applying dependency updates and reports the unresolved policy conflicts as findings.

Repository controls to add

Commit a project-level .npmrc with the exact policy baseline:

```
min-release-age=7
ignore-scripts=true
save-exact=true
package-lock=true
audit=true
fund=false
strict-peer-deps=true
```

Also add an exact packageManager field, SECURITY.md, dependency-update policy and CODEOWNERS protection for package.json, package-lock.json, .npmrc, workflows, Dockerfiles, deployment configuration and smart contracts.

Appendix A. Evidence map and references

Primary repository evidence

Topic	Path
Migration commitment and replay	contracts/src/LaunchpadMigration.sol
Harvest rush path	contracts/src/UniV2Migrator.sol
Curve buy/accounting	contracts/src/Launchpad.sol; LaunchpadBase.sol
LitVM deployment	contracts/script/DeployLitVM.s.sol; DeployTimelock.s.sol
Token swap flow	web/app/swap/page.tsx
Points persistence	web/lib/points/indexer.ts; web/lib/points/store.ts
Headers / SSRF controls	web/next.config.mjs; web/app/api/image/route.ts; web/netlify/edge-functions/image-rate-limit.ts
Dependency state	web/package.json; web/package-lock.json; netlify.toml
Operations	points/deploy/notus-points.service; ops/
Architecture / deployed rehearsal	README.md

External references

- Repository: <https://github.com/giupy997/launch-pad>
- Next.js SSRF advisory: <https://github.com/advisories/GHSA-cjq9-62q9-8jv4>
- ws memory exhaustion advisory: <https://github.com/advisories/GHSA-96hv-2xvq-fx4p>
- source-map-js denial-of-service advisory: <https://github.com/advisories/GHSA-68fv-2mvg-jv7q>
- nanoid infinite-loop advisory: <https://github.com/advisories/GHSA-2v37-7h3g-55p8>

Interpretation note

Line numbers in this report refer to the audited commit and may move after remediation. Candidate dependency versions are remediation starting points, not blanket approval: each exact new version still requires release-age, ownership, tarball, script, transitive, license, vulnerability and provenance review under the supplied policy.

Assessment conclusion

The project demonstrates serious attention to contract invariants and rehearsal operations. The review records four High, four Medium and three Low findings across migration trust, AMM execution, dependency security, deployment governance and operational controls. These findings and suggested remediations are presented for the project team's independent evaluation and prioritization; they do not state or imply a decision by LitVM.