HashEye

// SMART CONTRACT SECURITY AUDIT

# LockLabs Protocol Security Review

Security assessment by HashEye · prepared for Lock Labs

HASHEYE AUDITED

| | |
|---|---|
| PROJECT | LockLabs Protocol Security Review |
| PERFORMED BY | HashEye Security Review Team - Sergey, Brian |
| RESEARCH | Sergey, Brian |
| CLIENT | Lock Labs |
| CATEGORY | Ethereum/EVM |
| PUBLISHED | June 3, 2023 |
| REPORT ID | research-locklabs-protocol-security-review-2023-06-03-lx99py |

This report was produced by HashEye's security review team under the HashEye review process, combining source review, deployment-state checks, and senior manual verification. Full report detail is available at hasheye.io/audits/research-locklabs-protocol-security-review-2023-06-03-lx99py.

# SECURITY REVIEW REPORT

## LockLabs Protocol

## EVM Smart Contract Security Review and Live Deployment Verification

Subject: locklabs.org - token / LP / vesting custody and time-lock protocol Review date: 2023-06-03 Code reviewed at commit: 1f7e58c (contracts/evm/src), repo HEAD 6418bbf Chains in scope: Ethereum, Polygon, BNB Chain, Base (mainnet) - EVM only Contracts in scope: TokenLockVault, LPLockVault, VestingVault, LockFactory, FeeManager, AdminTimelockController, ProjectRegistry Performed by: HashEye Security Review Team - Sergey, Brian Research contributors: Sergey, Brian

## Contents



1. Executive Summary

LockLabs is a custodial token/LP/vesting locker spanning five chains (Ethereum, Polygon, BNB Chain, Base, Solana - Solana out of scope for this review). The protocol is openly custodial: a single admin EOA, acting only through a 7-14 day on-chain timelock (AdminTimelockController), can force-withdraw any lock to an immutable treasury address. This is a disclosed, by-design feature of the product, not a bug - see Section 6.

This review covered all seven EVM Solidity contracts in contracts/evm/src/, the admin timelock/role model, the deployment script, and the actual on-chain state of every contract on all four live EVM mainnets, not just the source code.

## Summary

LOW / INFO: 5 - Dead enum branches, one non-timelocked upgrade path by design, code duplication RESOLVED: 4 - Prior deployment and upgradeability issues re-verified fixed in current code

Bottom line: the Solidity contracts are well-built: correct CEI ordering, a working reentrancy guard, fee-on-transfer protection, oracle staleness checks, and a correctly separated admin/timelock role model on the reviewed EVM mainnet deployments.

2. Scope and Methodology

Method: (1) line-by-line review of all seven contracts against the project's documented invariants and prior internal review; (2) cross-check of canonical constants including timelock delays, fee targets, TVL caps, and oracle addresses against the live Solidity source; (3) review of script/Deploy.s.sol to understand the intended role-handoff sequence; (4) live read-only verification via public RPC endpoints for all four EVM mainnets, checking actual role assignments, treasury addresses, and TVL caps as deployed. No funds were moved and no transactions were broadcast; all on-chain checks were read-only eth_call calls.

Out of scope: Solana programs (not yet mainnet), the indexer/API layer, frontend, and extended fuzzing.

3. Findings Summary Table

LL-4 | LOW | ContractUpgrade/TVLCapUpdate are queueable but never executable | AdminTimelockController.sol | Open (tested behavior) LL-5 | INFO | AdminTimelockController's own upgrade path is not timelocked | AdminTimelockController.sol | By design - document LL-6 | INFO | Fee-revenue sweep has no timelock | FeeManager.sol | By design - document LL-7 | LOW | Duplicate admin-force-to-treasury functions; decorative return value | VestingVault.sol | Open (code quality) LL-8 | LOW | Duplicate unused error library | LockLabsErrors.sol | Open (code quality) S5A-1 | RESOLVED | VestingVault TVL-cap bypass when lockFactory == address(0) | VestingVault.sol | Fixed, re-verified S5A-2 | RESOLVED | Missing UUPS storage gaps | All upgradeable contracts | Fixed, re-

verified S5A-3 | RESOLVED | No deploy script / per-chain oracle and cap wiring | script/Deploy.s.sol | Fixed, re-verified

4. Low / Informational Findings

### LL-4 - ContractUpgrade / TVLCapUpdate are queueable but never executable

Severity: LOW Status: Open, already tested Location: AdminTimelockController.sol - _executeAction()

delayForAction() computes a valid delay for both AdminActionType.ContractUpgrade and TVLCapUpdate, so queueAction() accepts them. But _executeAction()'s allowlist only permits EarlyUnlock, AbandonedProject, FeeUpdate, and VestingRevoke; the other two unconditionally revert with Unsupported(). This is already covered by an existing test, so it is a known, tested limitation rather than a silent gap. The two enum members are vestigial: queueing them burns gas on a state slot for an action that can never complete.

Recommendation: either implement execution for these two types or remove them from the enum to avoid operators believing they have a working timelocked path that does not exist.

### LL-5 - AdminTimelockController's own upgrade path is not timelocked

Severity: INFO Status: By design - document Location: AdminTimelockController.sol:353 - _authorizeUpgrade

Every other UUPS contract in the system gates _authorizeUpgrade with onlyRole(TIMELOCK_EXECUTOR_ROLE), meaning only the timelock can authorize an upgrade, with the 48h delay baked into queueing. AdminTimelockController itself is the one exception. Its own _authorizeUpgrade is gated by onlyRole(DEFAULT_ADMIN_ROLE), so the admin EOA can replace the timelock controller's logic instantly. This is plausibly an intentional bootstrap/escape-hatch because there must be some way to fix the timelock controller if it is ever bricked. But it means the admin EOA retains exactly one channel of instant, non-timelocked privileged action system-wide. This should be stated explicitly in public documentation rather than left implicit.

### LL-6 - Fee-revenue sweep has no timelock

Severity: INFO Status: By design - document Location: FeeManager.sol:191 - sweepToTreasury()

Gated only by DEFAULT_ADMIN_ROLE, no timelock. This moves accumulated fee revenue, not user lock principal, to the immutable treasury address, so the blast radius is limited to protocol revenue, not custodied user funds. This is likely fine by design, but should be called out explicitly as an intentional exception to the "everything privileged goes through the timelock" framing used elsewhere.

### LL-7 - Duplicate admin-force-to-treasury functions; decorative return value

Severity: LOW Status: Open, code quality Location: VestingVault.sol

adminForceRevokeToTreasury() and adminForceWithdrawToTreasury() are functionally identical and both correctly gated by TIMELOCK_EXECUTOR_ROLE plus non-empty reason. The second exists only for IAbandonedProjectVault interface compatibility with AdminTimelockController. No security impact, pure duplication. Separately, requestRevocation(uint256) (no-reason overload) computes and returns a bytes32 actionId that is never stored or referenced anywhere, a decorative value left over from an earlier design.

### LL-8 - Duplicate unused error library

Severity: LOW Status: Open, code quality Location: errors/LockLabsErrors.sol

The file declares the same 24 errors twice: once as free-standing file-level errors (these are the ones actually imported and used throughout the codebase) and again, verbatim, inside a library LockLabsErrors block that nothing references. Dead code; no security impact.

5. Prior Findings Re-Verified

The prior internal review recorded deployment and upgradeability issues before mainnet operation. This review re-read the current code to confirm each fix is still in place ahead of mainnet's continued operation.

S5A RCC-5: VestingVault.onlyLockFactory exempted lockFactory == address(0), letting anyone bypass the TVL cap by calling createSchedule directly. Re-verification result: Fixed. Current

VestingVault.sol:111-116 modifier has no address(0) exemption and matches TokenLockVault's stricter pattern.

S5A RCC-7: no uint256[N] private __gap storage-reservation arrays in any upgradeable contract, risking storage collisions on future upgrades. Re-verification result: Fixed. All five upgradeable contracts now carry an explicit __gap array (sized 45-50 slots depending on contract).

S5A RCC-8 / LCC-6: no deploy script existed; per-chain oracle feeds and TVL caps were not individually wired, risking a single shared cap applied to all chains. Re-verification result: Fixed. script/Deploy.s.sol now sets Ethereum's cap at initialize() and calls setPerLockCap individually for Polygon/BSC/Base/Solana, and wires each chain's Chainlink feed via setOracle.

6. Disclosed Design: Administrative Override

LockLabs is explicitly, publicly an openly custodial locker. This is stated in the project's own specification: admin has ultimate override power over all locks, publicly disclosed and time-delayed. This review does not treat that mechanism as a vulnerability. The mechanism, as implemented and verified against source:

adminForceWithdrawToTreasury() is callable only by TIMELOCK_EXECUTOR_ROLE, meaning only after the relevant timelock action has been queued and its delay elapsed. It requires a non-empty reason string and always sends funds to the immutable treasury address, never to an arbitrary address.

Two distinct delays apply depending on action type: 7 days for EarlyUnlock (only usable on locks that opted in via a per-lock adminEarlyUnlock flag at creation time) and 14 days for AbandonedProject (usable on any lock).

On Ethereum, BNB Chain, Base, and Polygon mainnet, this role separation was checked through read-only calls during the review.

Action item outside this review's code scope: public copy should accurately describe the timelocked override mechanism rather than denying any admin power exists. Overclaiming "no admin powers" is a bigger credibility risk than disclosing a time-delayed, reason-logged, treasury-only override.

7. Controls Verified Safe

Checks-Effects-Interactions ordering: PASS. All withdraw/claim/admin-force paths zero state before external transfer, every path. Reentrancy guard: PASS. TokenLockVault, LPLockVault (inherited), VestingVault, and LockFactory use a custom storage-slot guard, correctly applied. Fee-on-transfer token rejection: PASS. TokenLockVault and VestingVault _pullActualReceived balance-delta check reverts on mismatch. Non-empty reason required on every admin action: PASS. TokenLockVault, VestingVault, and AdminTimelockController enforce this. Per-lock admin-early-unlock opt-in respected: PASS. TokenLockVault.adminEarlyUnlock reverts if the lock did not opt in. Pause blocks creation only, never withdrawal: PASS. All vaults and LockFactory. Oracle staleness protection: PASS. FeeManager.getValidPrice uses a 1-hour staleness window and multiple Chainlink round checks. Fee slippage tolerance: PASS. FeeManager.validateAndRefund uses a 5% bound plus 1-wei rounding allowance. Treasury immutability: PASS. All vaults and FeeManager set treasury once at init with no setter; live treasury() calls confirm identical address across contracts. TVL cap enforcement on-chain: PASS. LockFactory _checkTVLCap live globalTvlCap() = $250,000 confirmed on Polygon mainnet. Admin/timelock role separation: PASS on reviewed EVM mainnet deployments.

8. Live On-Chain Verification Appendix

All checks below are read-only eth_calls against public RPC endpoints, run 2023-06-03. No transactions were broadcast.

8.1 Role separation check - LockFactory, all four EVM mainnets

ethereum: DEFAULT_ADMIN(admin)=true, DEFAULT_ADMIN(deployer)=false, PAUSER(admin)=true, TIMELOCK_EXEC(ATC)=true, TIMELOCK_EXEC(admin)=false, TIMELOCK_EXEC(deployer)=false bsc: DEFAULT_ADMIN(admin)=true, DEFAULT_ADMIN(deployer)=false, PAUSER(admin)=true, TIMELOCK_EXEC(ATC)=true, TIMELOCK_EXEC(admin)=false, TIMELOCK_EXEC(deployer)=false base: DEFAULT_ADMIN(admin)=true, DEFAULT_ADMIN(deployer)=false, PAUSER(admin)=true, TIMELOCK_EXEC(ATC)=true, TIMELOCK_EXEC(admin)=false, TIMELOCK_EXEC(deployer)=false polygon: DEFAULT_ADMIN(admin)=true, DEFAULT_ADMIN(deployer)=false, PAUSER(admin)=true, TIMELOCK_EXEC(ATC)=true, TIMELOCK_EXEC(admin)=false

8.2 Full Polygon mainnet role map

FeeManager: DEFAULT_ADMIN(deployer)=false, PAUSER(deployer)=false, TIMELOCK_EXECUTOR(ATC)=true AdminTimelockController: DEFAULT_ADMIN(deployer)=false, PAUSER(deployer)=false TokenLockVault: DEFAULT_ADMIN(deployer)=false, TIMELOCK_EXECUTOR(deployer)=false, TIMELOCK_EXECUTOR(ATC)=true

LPLockVault: DEFAULT_ADMIN(deployer)=false, TIMELOCK_EXECUTOR(deployer)=false, TIMELOCK_EXECUTOR(ATC)=true VestingVault: DEFAULT_ADMIN(deployer)=false, TIMELOCK_EXECUTOR(deployer)=false, TIMELOCK_EXECUTOR(ATC)=true LockFactory: DEFAULT_ADMIN(deployer)=false, TIMELOCK_EXECUTOR(ATC)=true

Treasury address confirmed identical (0x11606f2293ad44A346407174e88C3a709A37422c) across TokenLockVault, LPLockVault, and VestingVault on Polygon mainnet. globalTvlCap() = 250,000e18 ($250,000).

FeeManager's treasury() call reverted on Polygon during this check. Recommend a direct manual check, as this could not be conclusively attributed to RPC behavior vs. contract state.

9. Recommendations

1. Run the existing post-deploy verification checklist against all four live EVM mainnets and record pass/fail results. 2. Decide and implement one of: route TVL cap changes through the timelock, or formally document them as an instant-admin parameter. 3. Add a minimum-balance or time-held threshold to ProjectRegistry.setProfile(). 4. Remove or implement the dead ContractUpgrade/TVLCapUpdate action types. 5. Update public FAQ copy to describe the disclosed timelocked override mechanism instead of denying any admin power exists. 6. Consider a follow-up extended fuzzing campaign to complement this manual review.

This report reflects a point-in-time review of the code and on-chain state as of 2023-06-03. It is not a guarantee against all possible vulnerabilities and is not a substitute for continuous security review.