Clober LiquidityVault

Security Review

HickupHH3

7 Feburary 2026

Contents

1 Introduction

The purpose of this audit is to review some minor changes made to the Clober LiquidityVault contract since the previous security review.

1.1 Audit Scope

The scope consisted of the `clober-liquidity-vault` repository in the `main` branch at commit hash `653880b76e8f5afef78381e803774d9c3ace7008`. The contracts found in the `src` folder that were included in scope were the following:

File
Operator.sol
LiquidityVault.sol
SimpleOracleStrategy.sol
oracle/ChainlinkOracle.sol
oracle/DatastreamOracle.sol

1.2 Audit Timeline

The audit was conducted from **6th Feburary** to **7th Feburary**.

1.3 Fix Review

A review of the fix was conducted subsequently on **7th April**.

1.4 Auditors Involved

HickupHH3

2 Risk Assessment Classification

There are 4 possible levels used to assess a vulnerability, with a separate section for gas optimizations.

High

Directly exploitable vulnerabilities with medium / high likelihood of loss of user funds, or contract functionality.

Resolving these issues are crucial to ensure the security and functionality of the contracts.

Medium

Vulnerabilities that relies on external dependencies / conditions to be met. Potentially leads to a loss of funds or functionality (eg. denial of service).

Resolving these issues are recommended to avoid undesired consequences.

Low

Issues arising from deviant behaviour than expected, but has no / little bearing from a security standpoint.

Informational

Issues that relate to security best practices recommendations, grammatical or styling errors, suggestions for variable/function name improvements etc. These issues are subjective and can be addressed based on the client's discretion.

While these issues may not directly affect the contract's functionality or security, addressing them can improve code readability, maintainability, and overall quality.

Gas Optimizations

Suggested changes to the codebase that will help reduce deployment or runtime gas costs, or to reduce the bytecode size should the limit be reached.

3 Findings Summary

Severity	No. of issues
High	0
Medium	0
Low	0
Informational	1
Gas Optimizations	0
Total	1

3.1 [Info] Uptime feed oracle needs to be updated to use `startedAt` instead of `updatedAt`

Context

ChainlinkOracle.sol#L127-L134

Details

Chainlink changed the behavior/spec of the uptime sequencer feed. Previously, the `updatedAt` parameter used to indicate when the sequencer feed last changed status. That has now been changed to the `startedAt` parameter.

Mitigation

Using the example consumer contract as reference, replace `updatedAt` with `startedAt` for the grace period check.

Response

Fixed at commit `9d96d92`.

Status

Fixed.

4 Disclaimer

The audit report provided reflects a thorough review conducted to the best of my ability. However, it is important to note that the time-boxing nature of the review and available resources may prevent the discovery of all potential security vulnerabilities. As such, this audit does not guarantee the absence of undiscovered vulnerabilities.

Furthermore, please be aware that the security review was conducted on a specific commit of the codebase, as indicated. Any subsequent modifications made to the code will necessitate a new security review to ensure comprehensive coverage.

Note that the contracts used in production and expected deployment values may defer significantly from what was reviewed.

To ensure a robust evaluation of the codebase, it is highly recommended to engage multiple auditors and firms, particularly for large and complex projects. The involvement of multiple perspectives can provide additional insights and potential missed vulnerabilities.

Please consider these factors when assessing the audit report and making decisions related to the security and reliability of the smart contracts. The security review is not an endorsement of the project or its team, and should not be treated as such.