Liquid Collective Review

PR #222 review

Auditors

Optimum, Lead Security Researcher

Xiaoming90, Security Researcher

Report prepared by: Pablo Misirov

August 7, 2023

Contents

1 About Spearbit

Spearbit is a decentralized network of expert security engineers offering reviews and other security related services to Web3 projects with the goal of creating a stronger ecosystem. Our network has experience on every part of the blockchain technology stack, including but not limited to protocol design, smart contracts and the Solidity compiler. Spearbit brings in untapped security talent by enabling expert freelance auditors seeking flexibility to work on interesting projects together.

Learn more about us at spearbit.com

2 Introduction

Liquid Collective is a multichain capable enterprise-grade liquid staking protocol, launching first on Ethereum. It allows institutional investors to stake and earn staking rewards while evidencing ownership of staked tokens in the form of a liquid staking token. Liquid Collective offers a solution that caters to the needs of institutions including:

- KYC / AML allowlisting process for all participants (including validators).
- Top performing node operators with multi-cloud, multi-region, and multi-client infrastructure.
- Governance by a broad and dispersed collective of industry participants.

Disclaimer: This pr review does not guarantee against a hack. It is a snapshot in time of PR 222 according to the specific commit. Any modifications to the code will require a new security review.

3 Risk classification

Severity level	**Impact: High**	**Impact: Medium**	**Impact: Low**
Likelihood: high	Critical	High	Medium
Likelihood: medium	High	Medium	Low
Likelihood: low	Medium	Low	Low

3.1 Impact

- High - leads to a loss of a significant portion (>10%) of assets in the protocol, or significant harm to a majority of users.
- Medium - global losses <10% or losses to only a subset of users, but still unacceptable.
- Low - losses will be annoying but bearable--applies to things like griefing attacks that can be easily repaired or even gas inefficiencies.

3.2 Likelihood

- High - almost certain to happen, easy to perform, or not easy but highly incentivized
- Medium - only conditionally possible or incentivized, but still relatively likely
- Low - requires stars to align, or little-to-no incentive

3.3 Action required for severity levels

- Critical - Must fix as soon as possible (if already deployed)
- High - Must fix (before deployment if not already deployed)
- Medium - Should fix
- Low - Could fix

4 Executive Summary

A change in the code base introduced by PR 222 has been reviewed by Optimum and Xiaoming. No security issues were found.

Summary

Project Name	liquid-collective
Repository	liquid-collective-protocol
Target PR	feature: ETH-951 deposit by multiples of 32 eth
Type of Project	Liquid Staking, DeFi
Review duration	2 days

Issues Found

Severity	**Count**
Critical Risk	0
High Risk	0
Medium Risk	0
Low Risk	0
Gas Optimizations	0
Informational	0
Total	**0**

5 Review result

After reviewing the PR 222, we don't see any security issues with this minor change or any risks that could negatively affect other parts of the codebase. Some observations:

- For the changes in Line 596, it simply changes the amount to commit each time to a multiple of 32 ETH, and the math in the change is correct and straightforward.
- As the final commit balance has to be in the multiple of 32 at the end, the initial commit balance must be of a multiple of 32. Thus, the `initRiverV1_2()` initialization function performs a rebalance between the "committed balance" and "balance to deposit" to achieve this, which is working as intended. The total asset (`_assetBalance()`) remains the same before and after the triggering of the new initialize function, so I don't see any negative side effect.
- `init` function looks fine - both from the normal init behavior and the fact that after it will be executed, any further change to `CommittedBalance` will always result in this value being a multiple of `DEPOSIT_SIZE`. Might be worth adding a short inline documentation for `initRiverV1_2`.