Cryptonics

Sifchain - Continuous Audit Report

Continuous Liquidity Pool (CLP)

January 31, 2021

Cryptonics Consulting S.L.
Ramiro de Maeztu 7
46022 Valencia
SPAIN

https://cryptonics.consulting/

Table of Contents

Disclaimer

THE CONTENT OF THIS AUDIT REPORT IS PROVIDED “AS IS”, WITHOUT REPRESENTATIONS AND WARRANTIES OF ANY KIND.

THE AUTHORS AND THEIR EMPLOYER DISCLAIM ANY LIABILITY FOR DAMAGE ARISING OUT OF, OR IN CONNECTION WITH, THIS AUDIT REPORT.

THIS AUDIT REPORT IS NOT A SECURITY WARRANTY, INVESTMENT ADVICE, OR AN ENDORSEMENT OF THE CLIENT OR ITS PRODUCTS. THIS AUDIT DOES NOT PROVIDE A SECURITY OR CORRECTNESS GUARANTEE OF THE AUDITED SOFTWARE.

Introduction

Purpose of this Report

Cryptonics Consulting has been engaged to perform a continuous security audit of Sifchain. This current audit report covers the implementation of the Sifchain **Continuous Liquidity Pool**..

The objectives of the audit are as follows:

1. Determine the correct functioning of the system, in accordance with the project specification.
2. Determine possible vulnerabilities, which could be exploited by an attacker.
3. Determine smart contract bugs, which might lead to unexpected behavior.
4. Analyze whether best practices have been applied during development.
5. Make recommendations to improve code safety and readability.

This report represents a summary of the findings.

As with any code audit, there is a limit to which vulnerabilities can be found, and unexpected execution paths may still be possible. The author of this report does not guarantee complete coverage (see disclaimer).

Codebase Submitted for the Audit

The audit has been performed on the code submitted in the following module in a public GitHub repository:

https://github.com/Sifchain/sifnode/tree/develop/x/clp

with the latest commit no: `df102a99a8dde6bdc7e512cef99d3eaae3ddbd0e`

The following files are in scope for the audit process:

```
x/clp
├── README.md
├── abci.go
├── alias.go
├── client
│   ├── cli
│   │   ├── flags.go
│   │   ├── query.go
```

```
│   │   ├── tx.go
│   │   └── tx_test.go
│   └── rest
│       ├── query.go
│       ├── rest.go
│       └── tx.go
├── clp_suite_test.go
├── genesis.go
├── genesis_test.go
├── handler.go
├── handler_test.go
├── keeper
│   ├── calculations.go
│   ├── executors.go
│   ├── keeper.go
│   ├── keeper_test.go
│   ├── liquidityprovider.go
│   ├── params.go
│   ├── pool.go
│   ├── querier.go
│   ├── querier_test.go
│   ├── validatorWhiteList_test.go
│   └── validatorWhitelist.go
├── module.go
└── types
    ├── asset.go
    ├── codec.go
    ├── errors.go
    ├── events.go
    ├── expected_keepers.go
    ├── genesis.go
    ├── keys.go
    ├── keys_test.go
    ├── msgs.go
    ├── msgs_test.go
    ├── params.go
    ├── querier.go
    └── types.go
```

Methodology

The audit has been performed by a mixed team of smart contract and full-stack auditors.

The following steps were performed:

1. Gaining an understanding of the code base's intended purpose by reading the available documentation.
2. Automated source code and dependency analysis.
3. Manual line by line analysis of the source code for security vulnerabilities and use of best practice guidelines, including but not limited to:
    a. Race condition analysis
    b. Under- / overflow issues
    c. Key management vulnerabilities
    d. Permissioning issues
    e. Logic errors
4. Report preparation

The results were then discussed between the auditors in a consensus meeting and integrated into this joint report.

Functionality Overview

The submitted code implements the continuous liquidity pool functionality built into the Sifchain protocol. Liquidity pools can be created for asset pairs, allowing swaps between Sifchain's native token (Rowan) and Sifchain representations of external tokens, providing the basic DEX facility offered by the protocol.

How to read this Report

This report classifies the issues found into the following severity categories:

Severity	Description
Critical	A serious and exploitable vulnerability that can lead to loss of funds, unrecoverable locked funds, or catastrophic denial of service.
Major	A vulnerability or bug that can affect the correct functioning of the system, lead to incorrect states or denial of service.
Minor	A violation of common best practices or incorrect usage of primitives, which may not currently have a major impact on security, but may do so in the future or introduce inefficiencies.
Informational	Comments and recommendations of design decisions or potential optimizations, that are not relevant to security. Their application may improve aspects, such as user experience or readability, but is not strictly necessary. This category may also include opinionated recommendations that the project team might not share.

The status of an issue can be one of the following: **Pending**, **Acknowledged** or **Resolved**. Informational notes do not have a status, since we consider them optional recommendations.

Note, that audits are an important step to improve the security of smart contracts and can find many issues. However, auditing complex codebases has its limits and a remaining risk is present (see disclaimer).

Users of the system should exercise caution. In order to help with the evaluation of the remaining risk, we provide a measure of the following key indicators: **code complexity**, **code readability**, **level of documentation**, and **test coverage**. We include a table with these criteria for each module, in the corresponding findings section.

Note, that high complexity or lower test coverage does not necessarily equate to a higher risk, although certain bugs are more easily detected in unit testing than a security audit and vice versa.

Detailed Findings

Notes on Architecture / Functionality

Apart from the code implementation security review, the audit team has performed a review of the currently supported functionality and architecture. The following a number of informal observations:

- **Pool addressing convention:**

  CLPs are identified using the symbol. According to the documentation the hash of the `pool-[firstTicker][secondTicker]` naming convention is used, even though only symbol key lookup can be evidenced in the code.

  In either case, this addressing scheme means that the token symbol is used (as throughout Sifchain) to identify individual assets. This could lead to collisions with assets using the same token symbols.

- **Gas Limitations:**

  The current implementation of the CLP seems to depend on Tendermint being configured to have an invite block gas limit, which seems to be the default. This is due to iterations over variable sized data structures. In particular, this is the case for destroying a liquidity pool, in which case the code iterates through all liquidity providers to refund funds. If a limit is placed on the amount of gas that can be consumed in a block, the transactions may fail if the number of liquidity providers is too large.

CLP Implementation

Summary of Findings

The Sifchain CLP component was found to contain no critical issues, no major issue, no minor issues, and 2 informational notes:

No	Description	Severity	Status
1	Leftover TODO comments	Informational	-
2	Documentation refers to Thorchain	Informational	-

Code Quality Criteria

Criteria	Status	Comment
Code complexity	Medium	-
Code readability and clarity	High	-
Level of Documentation	High	-
Test Coverage	Medium	-

Detailed Findings

1. Leftover TODO comments

Severity: Informational

In `abci.go`, no begin- or endblocks are defined and the empty functions are still marked with the auto-generated TODO comments.

Recommendation

Consider removing the functions for code clarity.

Status: Pending

2. Documentation refers to Thorchain

Severity: Informational

The CLP documentation refers to Thorchain instead of Sifchain throughout.

Recommendation

Consider updating the documentation.

Status: Pending

Appendix: Automated Code Analysis Report

```
[gosec] 2021/01/25 10:04:24 Including rules: default
[gosec] 2021/01/25 10:04:24 Excluding rules: default
[gosec] 2021/01/25 10:04:24 Import directory: /Users/stefan/audits/sifnode/x/clp
[gosec] 2021/01/25 10:04:51 Checking package: clp
[gosec] 2021/01/25 10:04:51 Checking file: /Users/stefan/audits/sifnode/x/clp/abci.go
[gosec] 2021/01/25 10:04:51 Checking file: /Users/stefan/audits/sifnode/x/clp/alias.go
[gosec] 2021/01/25 10:04:51 Checking file: /Users/stefan/audits/sifnode/x/clp/genesis.go
[gosec] 2021/01/25 10:04:51 Checking file: /Users/stefan/audits/sifnode/x/clp/handler.go
[gosec] 2021/01/25 10:04:51 Checking file: /Users/stefan/audits/sifnode/x/clp/module.go
Results:


Summary:
  Files: 5
  Lines: 644
  Nosec: 0
 Issues: 0
```