Introduction

This audit report highlights the overall security of the **liquid staking** smart contracts of the **Hipo Finance** project. With this report, I have tried to ensure the reliability of the smart contract by completing the assessment of their system's architecture and smart contract codebase.

For this audit, I completed a deep manual review of the code of the smart contract and all corresponding TypeScript files, including tests and wrappers.

Disclaimer

The audit does not give any warranties on the security of the code. One audit cannot be considered enough. I always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of the code.

Audit Details

- **Author:** Daniil Sedov
- **Date:** 13-29 October 2023
- **Languages:** FunC and TypeScript
- **Method:** Manual review

Scope of Audit

The focus of this audit was to verify whether the smart contract is secure, resilient, and working properly according to the specs.

Apart from reviewing the whole architecture of the system and its logic, I have completed a full review of the contract source code. The code review includes correctness, readability, quantity, and quality of test coverage.

Results

The audit uncovered a total of **4** issues, categorised as follows:

- **Critical:** Breakage of the whole system, theft of users' funds, and other similar vulnerabilities. These must be addressed before the release.
- **Major:** Breakage of some of the components, freezing of funds, and creating delays and lags in the system. These must be addressed before the release.

- **Medium:** Similar to the **Major** group, but problems from this group are much harder to reproduce in the real world for some reason. These must be addressed before the release.
- **Minor:** Small bugs that don't lead to any troubles, inefficient code, and bad practices. These are recommended to be addressed before the release.
- **Informational:** Issues related to code readability, naming, and other problems that don't affect the operation of the system. These are recommended to be addressed before the release.

Details for each of these categories are explained below.

Critical

(No critical issues found)

Major

(No major issues found)

Medium

(No medium issues found)

Minor

MIN-1

The JettonWallet smart contract, found in `wallet.fc`, allows for the withdrawal of surplus Toncoin from its balance. However, it does not offer the capability to withdraw stuck Jettons. For enhanced security, it's recommended to incorporate this feature. Kirill Emelyanenko's Modern Jetton can serve as a reference for the necessary additions.

Informational

INF-1

The majority of functions in `utils.fc` are marked with the `impure` specifier, which isn't appropriate. Only those functions that alter the contract's state should be designated as `impure`. Typically, these are functions that send messages or conduct checks that might result in throws.

INF-2

Within the `recv_internal` function in `treasury.fc`, there's a retrieval of a slice with 66 zero bits using `"00000000000000000"s.skip_bits(2)`. This action expends 26 gas for the `skip_bits` call. Given its constant nature, you can optimize this retrieval by using an asm function as follows:
`slice slice_66zerobits() asm "<b 0 66 u, b> <s PUSHSLICE";`

INF-3

The opcodes for the new operations appear to be miscalculated in relation to the established standards. To illustrate, the opcode for the `deposit_coins` operation ought to be determined using a CRC32 hash of `deposit_coins query_id:uint64 referrer:MsgAddress = InternalMsgBody`. This should be combined with a bitwise AND operation with `0x7fffffff`, yielding `0x103af428`. However, your TL-B schema displays `0x1375a585` instead.

Summary

No critical, major, or medium issues were identified, which indicates a relatively strong security stance. However, one minor and three informational issues were found.
Overall, the smart contracts were written in a well-structured manner, and all security-related aspects were well-considered. Additionally, the entire project is documented thoroughly and is fully covered with tests.