1inch Audit

Aggregation Router v4

September 2021

By CoinFabrik

Introduction 3
Summary 3
Contracts 3
Analyses 3
Issues Found by Severity 4
Enhancements 4
EN-01 Commented Code and Documentation 4
EN-02 Permitable Revert and Error Handling 4
Conclusion 4

Introduction

CoinFabrik was asked to audit the contracts for the 1inch Aggregation Router v4 which upgraded from version. 1inch Aggregation Router facilitates transactions by utilizing a wide range of protocols, and the audited contracts allow users to exchange tokens using other exchanges benefiting from the best rates at the moment.

No security-related issues were found. Next, we will describe the analyses performed.

Summary

The contracts audited are from the GitHub repository (link). The audit is based on the commit 0cdb810149b4750dbb3c857f3dabee794c313ca9.

Contracts

The audited contracts are:

- contracts/AggregationRouterV4.sol: facilitates trading across multiple DEX
- contracts/UnoswapV3Router.sol: facilitates swapping assets using the UniswapV3Pool
- contracts/ClipperRouter.sol: facilitates swapping assets using the ClipperRouter.
- contracts/LimitOrderProtocolRFQ.sol: fulfilling of RFQ orders.

Analyses

The following standard analyses were performed:

- Misuse of the different call methods
- Integer overflow errors
- Division by zero errors
- Outdated version of Solidity compiler
- Front running attacks
- Reentrancy attacks

- Misuse of block timestamps
- Softlock denial of service attacks
- Functions with excessive gas cost
- Missing or misused function qualifiers
- Needlessly complex code and contract interactions
- Poor or nonexistent error handling
- Failure to use a withdrawal pattern
- Insufficient validation of the input parameters
- Incorrect handling of cryptographic signatures

Furthermore, we looked for vulnerabilities that allow any of the parties in a swap or exchange to gain an undocumented advantage over other parties (including DEXes), vulnerabilities introduced by misuse of other contract's functions, and denial of service attacks against orders or swaps. In the case of the LimitOrderProtocolRFQ contract, we further looked for maliciously crafted orders and misuse of the order-filling functions finding no attacks.

Issues Found by Severity

Enhancements

EN-01 Commented Code and Documentation

It is recommended that functions and parameters are documented within the code, e.g., using the natspec format.

EN-02 Permitable Revert and Error Handling

The function `_permit()` in `contracts/helpers/Permitable.sol` follows that seems weak and it is not documented. In particular, the function emits an "Error" event in a case when it does not revert and will permit operations when the allowance is sufficient even if the permit failed. The logic and use cases should be documented.

Conclusion

We found the contracts to be mostly simple and straightforward. More importantly, we did not find any vulnerabilities. The code includes little documentation, though, and some enhancements in that direction were suggested.

MORE DETAILS

Disclaimer: This audit report is not a security warranty, investment advice, or an approval of the 1inch Aggregation Router v4 since CoinFabrik has not reviewed its platform. Moreover, it does not provide a smart contract code faultlessness guarantee.