soken

SMART CONTRACT SECURITY AUDIT

1inch (Farming)

Scan and check this report was posted at Soken Github

February, 2023

Website: soken.io

Table of Contents

Table of Contents 2
Disclaimer 3
Procedure 4
Terminology 5
Limitations 5
Basic Security Recommendation 5
Audit Scope 6
Audit Details 6
Social Profiles 7
Vulnerabilities checking 8
Security Issues 9
Soken Contact Info 11

Disclaimer

This is a comprehensive report based on our automated and manual examination of cybersecurity vulnerabilities and framework flaws of the project's smart contract.

Reading the full analysis report is essential to build your understanding of project's security level. It is crucial to take note, though we have done our best to perform this analysis and report, that you should not rely on the our research and cannot claim what it states or how we created it.

Before making any judgments, you have to conduct your own independent research.

We will discuss this in more depth in the following disclaimer - please read it fully.

DISCLAIMER: You agree to the terms of this disclaimer by reading this report or any portion thereof. Please stop reading this report and remove and delete any copies of this report that you download and/or print if you do not agree to these conditions. Scan and verify report's presence in the GitHub repository by a qr-code at the title page. This report is for non-reliability information only and does not represent investment advice. No one shall be entitled to depend on the report or its contents, and Soken and its affiliates shall not be held responsible to you or anyone else, nor shall Soken provide any guarantee or representation to any person with regard to the accuracy or integrity of the report.

Without any terms, warranties or other conditions other than as set forth in that exclusion and Soken excludes hereby all representations, warrants, conditions and other terms (including, without limitation, guarantees implied by the law of satisfactory quality, fitness for purposes and the use of reasonable care and skills).

The report is provided as "as is" and does not contain any terms and conditions. Except as legally banned, Soken disclaims all responsibility and responsibilities and no claim against Soken is made to any amount or type of loss or damages (without limitation, direct, indirect, special, punitive, consequential or pure economic loses or losses) that may be caused by you or any other person, or any damages or damages, including without limitations (whether innocent or negligent).

Security analysis is based only on the smart contracts. No applications or operations were reviewed for security. No product code has been reviewed.

Procedure

Our analysis contains following steps:

1. Project Analysis;

2. Manual analysis of smart contracts:
   - Deploying smart contracts on any of the network(Ropsten/Rinkeby) using Remix IDE
   - Hashes of all transaction will be recorded
   - Behaviour of functions and gas consumption is noted, as well.

3. Unit Testing:
   - Smart contract functions will be unit tested on multiple parameters and under multiple conditions to ensure that all paths of functions are functioning as intended.
   - In this phase intended behaviour of smart contract is verified.
   - In this phase, we would also ensure that smart contract functions are not consuming unnecessary gas.
   - Gas limits of functions will be verified in this stage.

4. Automated Testing:
   - Mythril
   - Oyente
   - Manticore
   - Solgraph

Terminology

We categorize the finding into 4 categories based on their vulnerability:

- Low-severity issue — less important, must be analyzed
- Medium-severity issue — important, needs to be analyzed and fixed
- High-severity issue —important, might cause vulnerabilities, must be analyzed and fixed
- Critical-severity issue —serious bug causes, must be analyzed and fixed.

Limitations

The security audit of Smart Contract cannot cover all vulnerabilities. Even if no vulnerabilities are detected in the audit, there is no guarantee that future smart contracts are safe. Smart contracts are in most cases safeguarded against specific sorts of attacks. In order to find as many flaws as possible, we carried out a comprehensive smart contract audit. Audit is a document that is not legally binding and guarantees nothing.

Basic Security Recommendation

Unlike hardware and paper wallets, hot wallets are connected to the internet and store private keys online, which exposes them to greater risk. If a company or an individual holds significant amounts of cryptocurrency in a hot wallet, they should consider using MultiSig addresses. Wallet security is enhanced when private keys are stored in different locations and are not controlled by a single entity.
More info: **https://blog.soken.io/how-to-gnosis-multisig-46b1386ba8e5**

Audit Scope

https://github.com/1inch/farming/tree/master/contracts:

- accounting/
  - FarmAccounting.sol
  - UserAccounting.sol

- interfaces/
  - IFarmingPod.sol
  - IFarmingPool.sol
  - IMultiFarmingPod.sol
- mocks/
  - EthTransferMock.sol
- /FarmingLib.sol
- /FarmingPod.sol
- /FarmingPool.sol
- /MultiFarmingPod.sol

Audit Details

Project Name: **1inch**

Language: **Solidity**

Compiler Version: **v0.8.0**

Social Profiles

Project Website: **https://aidefis.com/**

Project Twitter: **https://1inch.io/**

Project Telegram: **https://t.me/OneInchNetwork**

Project's Blog: **https://blog.1inch.io/**

Project's GitHub: **https://github.com/1inch**

Vulnerabilities checking

Issue Description	Checking Status
Compiler Errors	Completed
Delays in Data Delivery	Completed
Reentrancy	Low-issues
Transaction-Ordering Dependence	Completed
Timestamp Dependence	Completed
Shadowing State Variables	Completed
DoS with Failed Call	Completed
DoS with Block Gas Limit	Completed
Outdated Complier Version	Completed
Assert Violation	Completed
Use of Deprecated Solidity Functions	Completed
Integer Overflow and Underflow	Completed
Function Default Visibility	Completed
Malicious Event Log	Completed
Math Accuracy	Completed
Design Logic	Completed
Fallback Function Security	Completed
Cross-function Race Conditions	Completed
Safe Zeppelin Module	Completed

Security Issues

1) Fees for transfers are not checked: Medium-severity

FarmingPod.sol : Line #59

FarmingPool.sol : Line #62, #83

MultiFarmingPod.sol : Line #73

The argument **amount** should not be used as a trusted value in functions. This argument may not reflect the actual value being transferred (e.g. when there is a fee on transfer).

Recommendation:

Consider checking the actual amount received by the contract after the transfer function has been executed.

Status: Acknowledged.

2) Multiple Possible Reentrancy Points: Low-severity

FarmingPool.sol : Line #110

MultiFarmingPod.sol : Line #73

As the token variable can be any arbitrary token including a ERC-777 derived one, transfer hooks can be used to do reentrancy attacks.
The impact of both reentrancy points is low because in the FarmingPool.sol case, it can only be triggered by a distributor role.

Recommendation:

Refactor the function so it follows the check-effects-interaction pattern, or implement reentrancy guards.

Status:

All re-entrancies were fixed at commit:

bbb8b0db45dfb4ae0be860fef3ca4df9a761845f

Soken Contact Info

Website: www.soken.io

Mob: (+1)416-875-4174

32 Britain Street, Toronto, Ontario, Canada

Telegram: @team_soken

GitHub: sokenteam

Twitter: @soken_team