INTENTABLE

Powered by KIROBO

Intentable's Smart Transaction technology powered by the Kirobo FCT Platform

FCT Libraries Security Analysis

by Pessimistic

This report is public

October 2, 2024

Abstract

In this report, we consider the security of smart contracts of Intentable project. Our task is to find and describe security issues in the smart contracts of the platform.

Disclaimer

The audit does not give any warranties on the security of the code. A single audit cannot be considered enough. We always recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts. Besides, a security audit is not investment advice.

Summary

In this report, we considered the security of the FCT Libraries submodule of Intentable smart contracts. We described the audit process in the section below.

The report has designations like C01, M05, L10, N08. The letter represents severity, and the number represents the issue number.

This is a condensed version of the report; a more detailed one can be found at the following links:

- Report #1. The report includes the M04 issue, which was addressed and was relevant to the whole project.
- Report #2. The report includes the L04 issue, which was addressed and was relevant to the whole project.
- Report #3. The report includes:
    - M01, which was addressed and was relevant to the whole project;
    - L03, which was relevant to the current submodule.

The report includes the Unhandled ERC20 return values (M13 in the Report #2) issue of medium severity with `commented` status. All the tests passed. The code coverage is sufficient.

After the last recheck, which was fully described in Report #3 (see the Codebase update #8) and copied to the Project description, the codebase was updated again. The number of tests and the code coverage increased. We did not discover new issues.

After the recheck #1 in the current report, the codebase was updated. The developers added the new contract to the Libraries' scope, increased the code coverage, and fixed low severity issue.

We found one new issue of medium severity Incorrect values of constants, and two issues of low severity.

After the recheck #2, the codebase was updated again. The developers fixed Unhandled ERC20 return values and Incorrect values of constants issues of medium severity and all the low severity issues. The number of tests increased.

According to our recommendations, the developers split the long function in the **FCT_BatchMultiSig** contract, improving the code readability. However, it is still important to note that the project and its architecture are complex, though we realize that it is difficult or impossible to implement simply.

It is crucial to study the three reports above to get a full picture of the security of smart contracts.

General recommendations

We recommend implementing CI to run tests, calculate code coverage, and analyze code with linters and security tools.

Project overview

Project description

For the audit, we were provided with Intentable project on a private GitHub repository, commit cf31b27dd9e44e016ced8e3060a487e7e1e2d5cc.

It is a condensed version of the report. More detailed information about codebase updates can be found at the following links:

- Report #1;
- Report #2;
- Report #3.

The scope included:

- FCT_Lib_MultiCall.sol;
- FCT_Lib_UniswapV2.sol.

The documentation for the project included https://kirobo.gitbook.io/fct-developers-guide/.

The total LOC of the audited scope is 536.

All 592 tests passed. The code coverage of the project was 84.64%.

Codebase update #1

For the recheck, we were provided with Intentable project on a private GitHub repository, commit 9ba7299653676b06cadfcbfac2d1f9fff66d333c.

The developers provided the test results and the code coverage. All 630 tests passed, the code coverage of the scope was 97.61%, and the code coverage of the FCT platforms was 92.93%.

Codebase update #2

For the recheck, we were provided with Intentable project on a private GitHub repository, commit 78c90bb9bf2b38c50b03ef3d89f0d125e2bc3d6a.

The new **FCT_Lib_MultiCallV2.sol** contract was added to the scope, and its functionality description was added to the documentation.

The developers provided the test results and the code coverage. All 630 tests passed, the scope's code coverage was 98.25%, and the FCT platforms' code coverage was 95.67%.

Codebase update #3

For the recheck, we were provided with Intentable project on a private GitHub repository, commit 476e33239267aa680f46d7aec3312669fea7f9eb.

The developers fixed all medium and low severity issues and provided the test results and code coverage. All 634 tests passed, the scope's code coverage was 98.7%, and the FCT platforms' code coverage was 95.68%.

Audit process

We started to check the FCT platform around two years ago and created four extensive reports. The developers asked us to split these reports into seven submodules. Each submodule has:

- The last common commit for all reports;
- The same results of the tests and the code coverage for the whole project;
- The individual scope;
- Links to the corresponding previous reports;
- The list of fixed issues, e.g., M01, M04, L03 (see the description in the Summary), etc.;
- The descriptions of unfixed or commented issues that are still actual;
- The findings relevant to the whole project.

Issue descriptions copied from previous reports can have different contract names and lines, as they were checked again and updated due to the last commit in the Project description.

We started auditing the FCT platforms in 2022. During this time, we made three FCT reports and one report for the Smart Wallet part.

Each report is a continuation of the previous one. We made the split in the process to avoid one infinitely extensive report. The chronology of the reports:

- Smart Wallet report - finished on August 15, 2022;
- Report #1 - finished on November 17, 2022;
- Report #2 - finished on July 26, 2023, and last updated on January 16, 2024;
- Report #3 - finished on June 26, 2024.

See the previous reports for a more detailed description of the issues and process. The following rechecks related to a specific module will only appear in the corresponding report.

We made recheck #1 for this scope on June 26-27, 2024. It was the first recheck in this report but not the first one in the entire audit process (see the previous reports). The developers provided the test results and the code coverage, as we could not run them. This update included code refactoring and fixes for adding the new functionality. We did not find any new issues.

We made the recheck #2 from June 28 to July 2, 2024. We checked fixes for issues from the previous recheck and discovered M02 and low severity issues in the newly added **FCT_Lib_MultiCallV2** contract.

We scanned the project with the following tools:

- Static analyzer Slither;
- Our plugin Slitherin with an extended set of rules;
- Semgrep rules for smart contracts.

We asked the developers to provide the test results and the code coverage as we could not run them.

We combined in a private report all the verified issues we found during the manual audit or discovered by automated tools.

We made the recheck #3 on July 3, 2024. We checked whether the previous issues were fixed. We asked the developer for test results as we could not run them and recalculated the code coverage. Finally, we updated the report.

Manual analysis

The contracts were completely manually analyzed, their logic was checked. Besides, the results of the automated analysis were manually verified. All the confirmed issues are described below.

Critical issues

Critical issues seriously endanger project security. They can lead to loss of funds or other catastrophic consequences. The contracts should not be deployed before these issues are fixed.

The audit showed no critical issues.

Medium severity issues

Medium severity issues can influence project operation in the current implementation. Bugs, loss of potential income, and other non-critical failures fall into this category, as well as potential problems related to incorrect system management. We highly recommend addressing them.

M01. Unhandled ERC20 return values (fixed)

ERC-20 standard states: “Callers MUST handle false from returns (bool success). Callers MUST NOT assume that false is never returned!” However, the code in the **FCT_Lib_MultiCall** contract does not consistently check the returned value from the `approve` call in the `erc20Approvals` function.

The issue has been fixed and is not present in the latest version of the code.

M02. Incorrect values of constants (fixed)

The `value > INNER_RETX_MIN && value < INNER_RETX_REV_MIN` condition in the `FCT_Lib_MultiCallV2._replaceValue` function is never satisfied since `INNER_RETX_MIN = 0xEF000000..` and `INNER_RETX_REV_MIN = 0xEE000000...`.

It means that `INNER_RETX_MIN > INNER_RETX_REV_MIN`, which contradicts the initial condition.

The issue has been fixed and is not present in the latest version of the code.

Low severity issues

Low severity issues do not directly affect project operation. However, they might lead to various problems in future versions of the code. We recommend fixing them or explaining why the team has chosen a particular option.

L01. Type in the function name (fixed)

The `FCT_Lib_MultiCallV2._shouldReveryOnSuccess` function has a typo in the name. Consider replacing it with `_shouldRevert`**`t`**`OnSuccess`.

The issues have been fixed and are not present in the latest version of the code.

L02. Uncovered branches (fixed)

The code coverage is high enough, but it has uncovered branches in the `_shouldContinueOnSuccess`, `_shouldReveryOnSuccess`, `_shouldStopOnSuccess`, `_shouldContinueOnFail`, `_shouldStopOnFail`, `_shouldRevertOnFail` and `_replaceValue` functions of the **FCT_Lib_MultiCallV2** contract.

The issues have been fixed and are not present in the latest version of the code.

Notes

N01. Security precautions

Because the **FCT_Lib_MultiCallV2** contract is intended for everyone's use, it is important that no tokens are stored on the contract and no approvals or roles are given to the contract unless it is in the same transaction as multicall.

This analysis was performed by Pessimistic:

Daria Korepanova, Senior Security Engineer
Yhtyyar Sahatov, Security Engineer
Evgeny Bokarev, Junior Security Engineer
Konstantin Zherebtsov, Business Development Lead
Irina Vikhareva, Project Manager
Alexander Seleznev, CEO

October 2, 2024