



Powered by **KIROBO** 

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

FCT Flash Loans Executors Security Analysis

by Pessimistic

This report is public

October 2, 2024

Abstract	2
Disclaimer	2
Summary	2
General recommendations	3
Project overview	4
Project description	4
Audit process	5
Manual analysis	6
Critical issues	6
Medium severity issues	6
Low severity issues	6
Notes	6
N01. Non-zero balance after flashloan (commented)	6

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 Flash Loans Executors 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 fixed issues:
 - M10, L09, L13, N10, which were relevant to the current submodule;
 - L04, which was relevant for the whole project.
- [Report #3](#). The report includes:
 - N06, which was relevant to the current submodule;
 - M01, which was addressed and was relevant to the whole project.

The report includes the N01 (N08 in the [Report #2](#)) with the `commented` status. All the tests passed. The code coverage is sufficient.

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 the **FCT_FlashLoan_Aave_v2.sol** contract.

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

The total LOC of the audited scope is 195.

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

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.

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.

The audit showed no issues of medium severity.

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.

The audit showed no issues of low severity.

Notes

N01. Non-zero balance after flashloan (commented)

Any leftovers in the **FCT_FlashLoan_Aave_v2** contract balance after flashloan can be withdrawn by anyone via FCT to any address since `target` at line 105 can be any address.

Comment from the developers: We decided not to tract the leftovers in the contact (saves a lot of hassle and gas) and if there are leftovers in the contract they can be withdrawn by any FCT, we are aware and prefer it like that in order to keep the contract simple and to not have locked funds in the contract.

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