CPOOL Security Analysis

by Pessimistic

This report is public

21 September, 2021

Abstract

In this report, we consider the security of smarts contracts of CPOOL 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. One 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, security audit is not an investment advice.

Summary

In this report, we considered the security of CPOOL smart contracts. We performed our audit according to the procedure described below.

The audit showed only a few issues of low severity. They do not endanger project security.

After the initial audit, the code base was updated. All the issues were fixed.

General recommendations

We recommend adding NatSpec comments to the Vesting contract.

Project overview

Project description

For the audit, we were provided with CPOOL project on a public GitHub repository, commit fa83d98678c7ffe7d4705be8ed5657af5d0be537.

The project includes a README.md file and has a documentation.

The project compiles without any issues. All 13 tests pass, code coverage is unknown.

The total LOC of audited sources is 238.

Code base update

After the initial audit, the code base was updated. For the recheck, we were provided with commit d5ae1b235e3cdfd66573cfe5606500fd56ec78da.

Procedure

In our audit, we consider the following crucial features of the code:

1. Whether the code is secure.
2. Whether the code corresponds to the documentation (including whitepaper).
3. Whether the code meets best practices.

We perform our audit according to the following procedure:

- Automated analysis
  - We scan project's code base with automated tools: Slither and SmartCheck.
  - We manually verify (reject or confirm) all the issues found by tools.
- Manual audit
  - We manually analyze code base for security vulnerabilities.
  - We assess overall project structure and quality.
- Report
  - We reflect all the gathered information in 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 smart contracts security. We highly recommend fixing them.

The audit showed no critical issues

Medium severity issues

Medium issues can influence project operation in current implementation. We highly recommend addressing them.

The audit showed no medium severity issues

Low severity issues

Low severity issues can influence project operation in future versions of code. We recommend taking them into account.

Code quality (fixed)

- As `ABIEncoderV2` is the default option since solidity `0.8.0`, there is no need to enable it explicitly at line 4 of **CPOOL.sol**.
- Mark functions as `external` where possible in the **CPOOL** contract.

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

Gas consumption (fixed)

- Consider declaring `totalSupply` in **CPOOL** as a constant.
- Consider declaring `CPOOL` storage variable of the **Vesting** contract as `immutable`.

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

Notes

Potential contract misuse

Vesting contract would behave incorrectly if attempting to add more tokens via `holdTokens` when the claim process has been started already. It will require more tokens than actually needed, and those extra tokens would be stuck in the contract.

This analysis was performed by Pessimistic:

Evgeny Marchenko, Senior Security Engineer
Vladimir Tarasov, Security Engineer
Irina Vikhareva, Project Manager

21 September, 2021