Opium Security Analysis

by Pessimistic

This report is public.

Published: December 25, 2020

Abstract

In this report, we consider the security of staking smart contracts of Opium Network project. Our task is to find and describe security issues in 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 staking smart contracts of Opium Network project. We performed our audit according to the procedure described below.

The code is of high quality. The initial audit showed only a few issues of low severity. They do not endanger project security in any way.

After the audit, the code base was updated to the latest version. In this version, the developers added the documentation, also code quality issues were fixed.

General recommendations

We recommend moving to a modern Solidity version, as newer releases of the compiler have many improvements and optimizations.

Procedure

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

1. Whether code logic corresponds to the specification.
2. Whether the code is secure.
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: Crytic, MythX, and SmartCheck.
  - We manually verify (reject or confirm) all the issues found by tools.
- Manual audit
  - We inspect the specification and check whether the logic of smart contracts is consistent with it.
  - 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.

Project overview

Project description

In our analysis we consider staking smart contracts of Opium Network project on GitHub repository, commit cecb4af449005199e23afe037f87d3150f251681.

The total LOC of audited sources is 313.

Latest version of the code

After the initial audit, the code base was updated. For the recheck, we were provided with commit a1a3518f6c1af90d4c196d1ee76d30f26ce0f8eb.

The documentation was added to the project.

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 issues of medium severity.

Low severity issues

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

Code quality (fixed)

- In **OpiumStakingDerivatives** contract, `hedge()` function is `payable` for no reason.
- In **OpiumStakingDerivatives** contract at line 48, a visibility level is not declared for `premium` variable. Consider declaring it explicitly.
- Consider declaring functions as `external` instead of `public` where possible.

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

Compiler version

An old version of compiler is used in the project: `pragma solidity 0.5.16;`

We recommend upgrading to a newer version, e.g. `0.7.4`.

This analysis was performed by Pessimistic:

Evgeny Marchenko, Senior Security Engineer
Boris Nikashin, Analyst
Alexander Seleznev, Founder

December 25, 2020