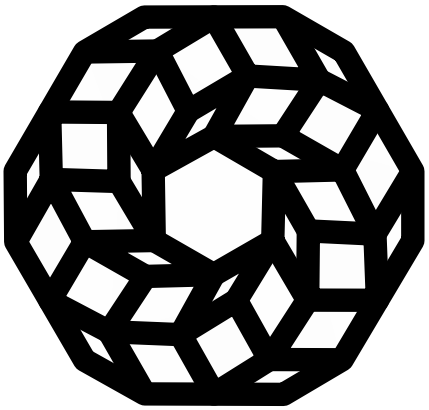

PowerPool PowerIndex SushiSwap Router Security Analysis

by Pessimistic

This report is public.

Published: April 27, 2021

Abstract

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

The analysis showed only one code quality issue of low severity.

The overall code quality is good. The project has the documentation and tests.

After the initial audit, the code base was updated. The update includes new functionality and a fix for the code quality issue. The audit of the updated code did not reveal any issues.

General recommendations

We do not have any further recommendations.

Project overview

Project description

In our analysis we consider PowerIndex SushiSwap Router smart contract of PowerPool project on GitHub repository, commit 8d386beef78490e55cb320807c07b93c1102ade5.

The scope of the audit included the analysis of the file **contracts/powerindex-router/implementations/SushiPowerIndexRouter.sol**.

For the audit, we were provided with the specification on GitHub repository, commit c00099b95545a9a1bcaed6e0d107bfead3f74628.

The total LOC of audited sources is 140.

Latest version of the code

After the initial audit, the code base was updated. For the recheck we were provided with commit 5f4b6f30010fecc2c82d23a41f6c5827cec5e88d.

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 tool 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.

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 `getUnderlyingBackedByXSushi()` function, consider replacing lines 121–125 with `getSushiForXSushi()` call.

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

This analysis was performed by Pessimistic:

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

April 27, 2021