Dragonfly in the Sun LLC EthGild Security Analysis

by Pessimistic

This report is public

16 November, 2021

Abstract

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

The audit discovered only one issue of low severity in the code. It does not endanger project security.

The main contract is implemented in a minimalist manner and relies on standards. The code base is heavily commented and well written.

General recommendations

We recommend addressing found issue and adding test coverage metrics to CI.

Project overview

Project description

For the audit, we were provided with EthGild project on a public GitHub repository, commit a60ede3bd8c207d74731c43a887d8fd59277b9c3. This smart contract has been deployed to ETH mainnet 0x10e79d0117865b48c825f7db7533ed619d68aac3.

The project is abundantly documented with many aspects, including security risk. Continuous integration involves running a static analyzer, which indicates a sound approach to potential threats.

All 16 tests pass, the code coverage is not measured.

The total LOC of audited sources is 110.

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 project security. They often lead to the loss of funds or other catastrophic failures. The contracts should not be deployed before these issues are fixed. We highly recommend fixing them.

The audit showed no critical issues

Medium severity issues

Medium issues can influence project operation in 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 medium severity issues

Low severity issues

Low severity issues do not directly affect project's operations. However, they might lead to various problems in the future versions of the code. We recommend taking them into account.

Code quality

Math operations will revert on over- and underflows by default starting with Solidity version `0.8.0`. It is recommended to remove SafeMath from the project to improve code readability and decrease gas consumption.

The developer tested gas consumption without SafeMath library (OpenZeppelin version 4.x). The difference on the project test suite (compiler 0.8.x optimized with 100 000 runs) is measured as 0.13-0.37%.

Notes

Locked ether

The contract contains a slow accumulation of locked ether for economic reasons.

This analysis was performed by Pessimistic:
Vladimir Tarasov, Security Engineer
Daria Korepanova, Security Engineer
Irina Vikhareva, Project Manager

16 November, 2021