P2P.ORG SECURITY AUDIT REPORT

April 10, 2023

MixBytes()

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Disclaimer

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only. The information presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of the Client. If you are not the intended recipient(s) of this document, please note that any disclosure, copying or dissemination of its content is strictly forbidden.

1.2 Security Assessment Methodology

A group of auditors are involved in the work on the audit. The security engineers check the provided source code independently of each other in accordance with the methodology described below:

1. Project architecture review:

- Project documentation review.
- General code review.
- Reverse research and study of the project architecture on the source code alone.

Stage goals

- Build an independent view of the project's architecture.
- Identifying logical flaws.

2. Checking the code in accordance with the vulnerabilities checklist:

- Manual code check for vulnerabilities listed on the Contractor's internal checklist. The Contractor's checklist is constantly updated based on the analysis of hacks, research, and audit of the clients' codes.
- Code check with the use of static analyzers (i.e Slither, Mythril, etc).

Stage goal

Eliminate typical vulnerabilities (e.g. reentrancy, gas limit, flash loan attacks etc.).

3. Checking the code for compliance with the desired security model:

- Detailed study of the project documentation.
- Examination of contracts tests.
- Examination of comments in code.
- Comparison of the desired model obtained during the study with the reversed view obtained during the blind audit.
- Exploits PoC development with the use of such programs as Brownie and Hardhat.

Stage goal

Detect inconsistencies with the desired model.

4. Consolidation of the auditors' interim reports into one:

- Cross check: each auditor reviews the reports of the others.
- Discussion of the issues found by the auditors.
- Issuance of an interim audit report.

Stage goals

- Double-check all the found issues to make sure they are relevant and the determined threat level is correct.
- Provide the Client with an interim report.

5. Bug fixing & re-audit:

- The Client either fixes the issues or provides comments on the issues found by the auditors. Feedback from the Customer must be received on every issue/bug so that the Contractor can assign them a status (either "fixed" or "acknowledged").
- Upon completion of the bug fixing, the auditors double-check each fix and assign it a specific status, providing a proof link to the fix.
- A re-audited report is issued.

Stage goals

- Verify the fixed code version with all the recommendations and its statuses.
- Provide the Client with a re-audited report.

6. Final code verification and issuance of a public audit report:

- The Customer deploys the re-audited source code on the mainnet.
- The Contractor verifies the deployed code with the re-audited version and checks them for compliance.
- If the versions of the code match, the Contractor issues a public audit report.

Stage goals

- Conduct the final check of the code deployed on the mainnet.
- Provide the Customer with a public audit report.

Finding Severity breakdown

All vulnerabilities discovered during the audit are classified based on their potential severity and have the following classification:

Severity	Description
Critical	Bugs leading to assets theft, fund access locking, or any other loss of funds.
High	Bugs that can trigger a contract failure. Further recovery is possible only by manual modification of the contract state or replacement.
Medium	Bugs that can break the intended contract logic or expose it to DoS attacks, but do not cause direct loss funds.
Low	Bugs that do not have a significant immediate impact and could be easily fixed.

Based on the feedback received from the Customer regarding the list of findings discovered by the Contractor, they are assigned the following statuses:

Status	Description
Fixed	Recommended fixes have been made to the project code and no longer affect its security.
Acknowledged	The Customer is aware of the finding. Recommendations for the finding are planned to be resolved in the future.

1.3 Project Overview

The audited scope contains several smart contracts, designed to deposit ETH to validators and to distribute staking rewards among the depositor, the service, and the referrer who attracted the depositor to the project.

`Oracle.sol` implements the export of off-chain data (CL rewards amount) using the Merkle Tree technique.

`P2pEth2Depositor.sol` implements batch deposit of ETH into multiple validators using suppiled credentials.

`FeeDistributorFactory.sol` is a `Factory` contract that performs deployment of fee distributor contracts using the `EIP 1167` technique.

`FeeDistributor.sol` implements the rewards distribution.

1.4 Project Dashboard

Project Summary

Title	Description
Client	P2P.ORG (P2P Staking)
Project name	ETH2 Depositor & ETH Staking Fee Distributor
Timeline	30 Mar 2023 - 6 Apr 2023
Number of Auditors	2

Project Log

Date	Commit Hash	Note
30.03.2023	5a8c0165e56abda4bd3946c4c0906170b0fc9039	initial commit
05.04.2023	c65caf5637c2560dcaf53918f6f7471d93a572f5	audit with fixes
07.04.2023	c65caf5637c2560dcaf53918f6f7471d93a572f5	verified deployments by bytecode exact match

Project Scope

The audit covered the following files:

File name	Link
contracts/access/Ownable2Step.sol	Ownable2Step.sol
contracts/access/OwnableBase.sol	OwnableBase.sol

File name	Link
contracts/access/Ownable.sol	Ownable.sol
contracts/access/OwnableWithOperator.sol	OwnableWithOperator.sol
contracts/assetRecovering/AssetRecoverer.sol	AssetRecoverer.sol
contracts/assetRecovering/OwnableAssetRecoverer.sol	OwnableAssetRecoverer.sol
contracts/assetRecovering/OwnableTokenRecoverer.sol	OwnableTokenRecoverer.sol
contracts/assetRecovering/TokenRecoverer.sol	TokenRecoverer.sol
contracts/feeDistributor/FeeDistributor.sol	FeeDistributor.sol
contracts/feeDistributorFactory/FeeDistributorFactory.sol	FeeDistributorFactory.sol
contracts/oracle/Oracle.sol	Oracle.sol
contracts/p2pEth2Depositor/P2pEth2Depositor.sol	P2pEth2Depositor.sol
contracts/p2pMessageSender/P2pMessageSender.sol	P2pMessageSender.sol

Deployments

Contract	Address	tx hash
FeeDistributorFactory	0xd5B7680f95c5A6CAeCdBBEB1DeE580960C4F891b	0x713e6dc704e173d5f1451f5414f1f6ddd703a57fb6ae632622415 4d783c415d4
Oracle	0x105D2F6C358d185d1D81a73c1F76a75a2Cc500ed	0x990cc4fd7b61dab9f9a540714031f11fa5ee9fe237d8c50ec878edb02f867662
P2pEth2Depositor	0x8e76a33f1aFf7EB15DE832810506814aF4789536	0x48d27f70301019cf6703a8766e8112f04a481f918318a30b464cb8fafd08dc33
FeeDistributor	0x5025B68b079149424c9102d3978f4FcC4aC4FFEC	0x67bba198babe3fbecf7d5760093a76b7d63a3a3c69e376032e6806537bd7b447

1.5 Summary of findings

Severity	# of Findings
Critical	0
High	0
Medium	0
Low	1

ID	Name	Severity	Status
L-1	The Oracle can lock reward distribution	Low	Fixed

1.6 Conclusion

During the audit process, 1 low severity issue has been found and fixed by the developers.

2.FINDINGS REPORT

2.1 Critical

Not Found

2.2 High

Not Found

2.3 Medium

Not Found

2.4 Low

L-1	The Oracle can lock reward distribution
Severity	Low
Status	Fixed in c65caf56

Description

If Oracle updates become unavailable for some reasons, the rewards cannot be distributed.

Recommendation

It is recommended to introduce a function that can distribute rewards without data from the Oracle.

3. ABOUT MIXBYTES

MixBytes is a team of blockchain developers, auditors and analysts keen on decentralized systems. We build opensource solutions, smart contracts and blockchain protocols, perform security audits, work on benchmarking and software testing solutions, do research and tech consultancy.

Contacts

https://github.com/mixbytes/audits_public

https://mixbytes.io/

hello@mixbytes.io

https://twitter.com/mixbytes