MixBytes()

EMPOWER THE DAO

SMART CONTRACT
AUDIT REPORT

OCTOBER 24
2019

FOREWORD TO REPORT

A small bug can cost you millions. **MixBytes** is a team of experienced blockchain engineers that reviews your codebase and helps you avoid potential heavy losses. More than 10 years of expertise in information security and high-load services and 15 000+ lines of audited code speak for themselves. This document outlines our methodology, scope of work, and results. We would like to thank **Empower the Dao** for their trust and opportunity to audit their smart contracts.

CONTENT DISCLAIMER

This report is public upon the consent of **Empower the Dao**. **MixBytes** is not to be held responsible for any damage arising from or connected with the report. Smart contract security audit does not guarantee an inclusive analysis disclosing all possible errors and vulnerabilities but covers the majority of issues that represent threat to smart contract operation, have been overlooked or should be fixed.

TABLE OF CONTENTS

01 INTRODUCTION TO THE AUDIT

GENERAL PROVISIONS

Empower the DAO team is working on integrating Aragon with a number of well-known Ethereum-based projects and their communities, aiming to deliver real business value for end users.

With this in mind, **MixBytes** team is willing to contribute to **Empower the DAO** initiatives by providing security assessment of the Compound, Uniswap and ENS smart contracts and their dependencies.

SCOPE OF THE AUDIT

The scope of the audit included:

1. **The Compound contract**
2. **The Compound contract dependency**
3. **The Uniswap contract**
4. **The Uniswap contract dependency**
5. **The EnsApp contract**

02 SECURITY ASSESSMENT PRINCIPLES

CLASSIFICATION OF ISSUES

CRITICAL

Bugs leading to Ether or token theft, fund access locking or any other loss of Ether/tokens to be transferred to any party (for example, dividends).

MAJOR

Bugs that can trigger a contract failure. Further recovery is possible only by manual modification of the contract state or replacement.

WARNINGS

Bugs that can break the intended contract logic or expose it to DoS attacks.

COMMENTS

Other issues and recommendations reported to/acknowledged by the team.

SECURITY ASSESSMENT METHODOLOGY

The audit was performed by 2 auditors. Stages of the audit were as follows:

1. “Blind” manual check of the code and its model
2. “Guided” manual code review
3. Checking the code compliance with customer requirements
4. Automated security analysis using the internal solidity security checker
5. Automated security analysis using public analyzers
6. Manual checklist system inspection
7. Discussion of independent audit results
8. Report preparation

03 DETECTED ISSUES

CRITICAL

Not found.

MAJOR

Not found.

WARNINGS

1. Compound.sol#L131
Uniswap.sol#L143

It is possible to transfer more Ether to the `agent` balance than it was transferred via the `deposit` call (`msg.value`). Moreover, the access to this feature is not limited in any way. We recommend that you prohibit this behavior and require `_value == msg.value` in the case of `_token == ETH`.

Status:

FIXED at **ebef91a42914cfaee8e64423306ce8c7d3157b3e**
and **0c097247cbed0aaaba666ef99df346b7b3cb3b7d**

2. Compound.sol#L87

After this call tokens that were minted using the previous `agent` will be unavailable. We suggest making sure that all the tokens minted using the previous `agent` were redeemed.

Status:

FIXED at **ebef91a42914cfaee8e64423306ce8c7d3157b3e**

3. Compound.sol#L112

After this call tokens that were minted using _cErc20 will be unavailable. We advise verifying that all the token minted via the passed _cErc20 were redeemed.

Status:

FIXED at **ebef91a42914cfaee8e64423306ce8c7d3157b3e**

4. Uniswap.sol#L202

If the tokenToEthSwapInput call result is less than the _minEthAmount value, annule the token approval or roll back the transaction. Therefore, if the deal fails, the exchange contract will not be able to withdraw tokens afterwards. The Uniswap documentation does not state that the transaction will be rolled back in case of a failed deal.

Status:

FIXED at **0c097247cbed0aaaba666ef99df346b7b3cb3b7d**

COMMENTS

1. Compound.sol#L129

Supporting Ether transfer to the agent is irrelevant as working with CEther is not supported.

Status:

ACKNOWLEDGED

2. Compound.sol#L68-L70
Compound.sol#L97-L101

General token validation code (and Agent, perhaps) should be moved to a separate internal method.

Status:

ACKNOWLEDGED

3. Compound.sol#L166

Compound.sol#L179

We suggest adding informative parameters to the events.

Status:

FIXED at **ebef91a42914cfaee8e64423306ce8c7d3157b3e**

4. General code

The `deposit`, `transfer`, and the `enabledTokens` array control functions can be moved to a `compound-aragon-app` and `uniswap-aragon-app` base contract.

Status:

ACKNOWLEDGED

5. EnsApp.sol#L76

The comment must have been copied from the `setAgent` function and should be corrected.

Status:

FIXED at **cb28347db70c830485a4405fea2eaf2b10067780**

6. Compound.sol#L119

Compound.sol#L213

Uniswap.sol#L131

We recommend adding the `isInitialized` modifier.

Status:

FIXED at **ebef91a42914cfaee8e64423306ce8c7d3157b3e**
and **0c097247cbed0aaaba666ef99df346b7b3cb3b7d**

7. Uniswap.sol#L68

Uniswap.sol#L69

The `ERROR_NOT_CONTRACT` revert reason is not informative enough as it is unclear to which address it is related. We suggest creating separate revert reasons for `Agent` and `UniswapFactoryInterface`.

Status:

FIXED at **0c097247cbed0aaaba666ef99df346b7b3cb3b7d**

8. **Uniswap.sol#L175**
   Uniswap.sol#L196

Exchange check should be moved to a modifier.

Status:

FIXED at **0c097247cbed0aaaba666ef99df346b7b3cb3b7d**

9. **Uniswap.sol#L34**
   Compound.sol#L32

The constant is not used and can be removed.

Status:

FIXED at **0c097247cbed0aaaba666ef99df346b7b3cb3b7d**
and **ebef91a42914cfaee8e64423306ce8c7d3157b3e**

04 CONCLUSION AND RESULTS

Overall code quality is above average. Attention must be paid to excessive Ether transfer to the Agent and temporary access lock to the funds sent to Compound. Also, code support can be facilitated by moving the general code to a base contract.

The contracts:

1. **Compound**
2. **Compound dependency**
3. **Uniswap**
4. **Uniswap dependency**
5. **EnsApp**

don’t have any vulnerabilities according to our analysis.

ABOUT MIXBYTES

MixBytes is a team of blockchain developers, auditors and analysts keen on decentralized systems. We build open-source solutions, smart contracts and blockchain protocols, perform security audits, work on benchmarking and software testing solutions, consult universities and enterprises, do research, publish articles and documentation.

Stack

Blockchains

JOIN US