MixBytes()

CURVE VOTING

(ARAGON VOTING FORK)

SMART CONTRACT AUDIT REPORT

JULY 22
2020

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 18 000+ lines of audited code speak for themselves. This document outlines our methodology, scope of work, and results. We would like to thank **Curve Finance** for their trust and opportunity to audit their smart contracts.

CONTENT DISCLAIMER

This report is public upon the consent of **Curve Finance**. **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

Curve Finance is a project that uses liquidity pools and bonding curves to provide high-efficiency stablecoin trading and low-risk returns for liquidity providers. **MixBytes** was approached by **Curve Finance** to provide a security assessment of a part of their governance mechanism smart contracts.

SCOPE OF THE AUDIT

The scope of the audit is smart contracts at **https://github.com/pengiundev/curve-aragon-voting/blob/a988e7c9a0543b58f0f0adb93a7b06acd5b36c0c/contracts/Voting.sol**

Audited commit is a988e7c9a0543b58f0f0adb93a7b06acd5b36c0c.

The specification of the contract is located **https://github.com/pengiundev/curve-aragon-voting/blob/e1d824768dca1a69d7dfc1570e72828d91466e05/SPECS.md**

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 with triple redundancy by three 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 to customer requirements
4. Discussion of independent audit results
5. Report preparation

03 DETECTED ISSUES

CRITICAL

Not found.

MAJOR

1. Voting.sol#L254

`canCreateNewVote` is not taken into account here, although it can block new vote creation. Which, in turn, will break proper forwarding in the Aragon UI. We suggest adding `&& canCreateNewVote()` here.

Status:

FIXED at **757970c**

WARNINGS

1. Voting.sol#L60-L62

Global variables initialized in such a way won’t work in aragon apps. **Explanation**.

We recommend setting them in `initialize` function like this:

```
function initialize() .. {
     ...
minBalanceLowerLimit = <value literal>;
minTimeLowerLimit = <value literal>;
minTimeUpperLimit = <value literal>;
...
}
```

Status:

FIXED at **Voting.sol#L147-L149**

2. Voting.sol#L125-L126

The values are not checked against the limits. We suggest adding extra checks.

Status:

FIXED at **a92bf07**

3. Voting.sol#L60

Looks like it's assumed that the token's decimals are 18, which is not always the case. We suggest calculating this value dynamically taking into account `token.decimals`.

Status:

FIXED at **757970c**

4. Voting.sol#L162-L174

We recommend adding Radspec documentation to the functions. Otherwise, users of Aragon UI will have troubles calling them.

Status:

FIXED at **48da227**

COMMENTS

1. Voting.sol#L20-L25

Solidity constants are not optimized. They work like pure functions, executed upon each access. We suggest using the same pattern as here: **BalanceTimeForwarder.sol#L12**.

Status:

FIXED at **bf14d4b**

2. Voting.sol#L103

We recommend adding docs on `_minTime` and `_minBalance` parameters

Status:

FIXED at **48da227**

3. **Voting.sol#L88**
   Voting.sol#L94
   Voting.sol#L163
   Voting.sol#L170

The comments and the error message might not be consistent with the actual constraint values.

Status:

FIXED at **48da227**

4. **Voting.sol#L279**

The function should be marked as `view`.

Status:

FIXED at **48da227**

04 CONCLUSION AND RESULTS

We find the implemented incentivisation approach as a clever solution to the lingering voters problem.

Several troublesome issues were identified and properly addressed.

The **fixed contract** doesn’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