Voting Connectors Smart Contracts Audit Report

Introduction

Aragon is software allowing to freely organize and collaborate without borders or intermediaries.

Aragon One is a Swiss company formed by the founders of the Aragon project, building the tools and community necessary for the project to succeed.

Voting Connectors are apps that serve as bridges to Aragon Voting apps requiring checkpointed balances (or any other app that requires checkpointed balances).

- Token Wrapper: wrap external tokens to a checkpointed token.
- Voting Aggregator: aggregate voting power over multiple sources.

With this in mind, MixBytes team was willing to contribute to Aragon ecosystem development by providing security assessment of the Voting Connectors smart contracts.

Scope of the audit

The scope of the audit included:

- contract utils version ae01814 (except the `test` subdirectory)
- TokenWrapper.sol version ae01814
- VotingAggregator.sol version ae01814

Security Assessment Principles

Classification of Issues

- CRITICAL: Bugs that enable theft of ether/tokens, lock access to funds without possibility to restore it, or lead to any other loss of ether/tokens to be transferred to any party (for example, dividends).
- MAJOR: Bugs that can trigger a contract failure, with further recovery only possible through manual modification of the contract state or contract replacement altogether.
- WARNINGS: Bugs that can break the intended contract logic or enable a DoS attack on the contract.
- COMMENTS: All other issues and recommendations.

Security Assessment Methodology

The audit was performed with triple redundancy by three auditors.

Stages of the audit were as follows:

- “Blind” manual check of the code and model behind the code
- “Guided” manual check of the code
- Check of adherence of the code to requirements of the client
- Automated security analysis using internal solidity security checker
- Automated security analysis using public analysers
- Manual by-checklist inspection of the system
- Discussion and merge of independent audit results
- Report execution

Detected Issues

CRITICAL

Not found

MAJOR

1. VotingAggregator.sol#L299

Power source weight is not checkpointed, that makes vote manipulation possible. The issue was identified by the client after examining the intermediary audit report.

Fixed at c25f24f

WARNINGS

1. VotingAggregator.sol#L291

An unbound loop with external calls can have high gas consumption. As a result, block gas limit may prevent some transactions from being executed. We recommend adding a limit to the source number.

Fixed at 39c6cca

2. VotingAggregator.sol#L131

`_weight` can be set to zero. This check implies that such behavior is unfavourable. We suggest adding a similar check to the `changeSourceWeight` function.

Fixed at f31c35f

COMMENTS

1. ActivePeriod.sol#L78

We suggest adding a check that a period with a given index exists.

Deleted (`ActivePeriod` was removed)

2. Checkpointing.sol#L33

ActivePeriod.sol#L36

ActivePeriod.sol#L56

APIs of the `Checkpointing` and `ActivePeriod` libraries can be made more explicit in terms of the supported data types (`uint64` for time-like values and `uint192` for numeric values). We suggest using exact data types and forcing users of the libraries to acknowledge that by using type casts. Interestingly enough, there is a ready-made `getBlockNumber64` function, which perfectly fits into the picture.

Fixed at 935259d

3. TokenWrapper.sol#L87

We recommend adding a warning to the documentation of the `TokenWrapper` contract, stating that neither `totalSupply` nor any balance of the token can exceed the `MAX_UINT192` value.

Fixed at 8d0506c

4. VotingAggregator.sol#L271

Typo in the word `activation` .

Deleted (the method was removed)

5. VotingAggregator.sol#L103

Many power sources with the same address can be added. Make sure that this is the expected scenario.

Fixed be88283

6. VotingAggregator.sol#L131

The function can be executed even for a disabled power source. Make sure that this is the desired behavior.

Acknowledged

7. VotingAggregator.sol#L297

The `_aggregateAt` function can be temporarily blocked by a malicious power source.

Extra checks added at 4d9da90

8. VotingAggregator.sol#L325

ActivePeriod.sol#L128

We recommend using `assert` instead of `revert` here, since it is a better way to check the code consistency.

Acknowledged

CONCLUSION

Overall code quality is high. In the course of our analysis we found only a couple of minor slips, several comments and suggestions were made.

The client identified a major issue after examining the intermediary audit report. The issue was addressed and fixed properly.

The fixed contracts don't have any vulnerabilities according to our analysis.