COINMERCENARY / SMART CONTRACT AUDIT

ARAW Token

20 MAY 2018 / TABLE OF CONTENTS

INTRODUCTION

CoinMercenary provides comprehensive, independent smart contract auditing.

We help stakeholders confirm the quality and security of their smart contracts using our comprehensive and standardized audit process. Each audit is unbiased and verified by multiple reputable auditors.

The scope of this audit was to analyze and document the ARAW token generation event contract.

This audit provides practical assurance of the logic and implementation of the contract.

AUDIT METHODOLOGY

CoinMercenary audits consist of four categories of analysis.

Design Patterns

We first inspect the overall structure of the smart contract, including both manual and automated analysis.

The design pattern analysis checks appropriate test coverage, utilizes a linter to ensure consistent style and composition, and code comments are reviewed. Overall architecture and safe usage of third party smart contracts are checked to ensure the contract is structured in a way that will not result in future issues.

Static Analysis

The static analysis portion of our audit is performed using a series of automated tools, purposefully designed to test the security of the contract. These tools include:

- **Manticore** - Dynamic binary analysis tool with EVM support.
- **Mythril** - Reversing and bug hunting framework for the Ethereum blockchain.
- **Oyente** - Analyzes Solidity code to find common vulnerabilities.
- **Solgraph** - DOT graph creation for visualizing function control flow of a Solidity contract to highlight potential security vulnerabilities.

Data flow and control flow are also analyzed to identify vulnerabilities.

Manual Analysis

Performing a hands on review of the smart contract to identify common vulnerabilities is the most intensive portion of our audit. Checks for race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks are part of our standardized process.

Network Behavior

In addition to our design pattern check, we also specifically look at network behavior. We model how the smart contract will operate once in production,

then determine the answers to questions such as: how much gas will be used, are there any optimizations, how will the contract interact?

Contracts Reviewed

On May 20, 2018 using git hash 09431576baac629e578cae4505a2309e0496fc38 the following contract files and their respective SHA256 fingerprints were reviewed:

Filename	SHA256 Fingerprint
ArawToken.sol	9aaa022e7e0eb4242b2f836eefe1b485fe94ed5fb256968ad224cbf7dfa65ded
BasicToken.sol	c77fd79df02bab6a41bcba3114216a964a9ceb2ac8747fe52bb358935d34c8f7
TimeLock.sol	8f0f15a5ba979228a3fa43bd0692b904e1b142870bf45444d673adb1fa7c204f

Remediation Audit

On May 28, 2018 using git hash 68b933810ff0d0494264f9660abd4c9ccd69d63f the following contract files and their respective SHA256 fingerprints were reviewed:

Filename	SHA256 Fingerprint
ArawToken.sol	f42a42a91e6ce54f3823edd765071b560c2185c6bc418dfedb152112b320e1bd
BasicToken.sol	c77fd79df02bab6a41bcba3114216a964a9ceb2ac8747fe52bb358935d34c8f7

Remediation Audit (Final)

On June 15, 2018 using git hash 628feabde7e145655533b64e6037fbf6839fe15a the following contract files and their respective SHA256 fingerprints were reviewed:

Filename	SHA256 Fingerprint
ArawToken.sol	f42a42a91e6ce54f3823edd765071b560c2185c6bc418dfedb152112b320e1bd
AirDrop.sol	9f4138e7db622cc93f8f6b2928daf283c6bc8c0eb3d47351c8a562f6ad9833a2
PrivateSale.sol	ba3d50576914131b68cf12c24440bfddd6234b0716873d96d7601eace0f968a9

Contract Deployment Address

Contract Deployment Address: 0x30680AC0a8A993088223925265fD7a76bEb87E7F

AUDIT SUMMARY

The contracts have been found to be free of security issues.

Analysis Results

	Initial Audit	**Remediation Audit**
Design Patterns	**Failed**	Passed
Static Analysis	Passed	Passed
Manual Analysis	**Failed**	Passed
Token Allocation	Passed	Passed
Network Behavior	Passed	Passed

Test Results

- Extensive test coverage available.

Token Allocation Results

- Symbol: ARAW
- Locking contract available.
- 5,000,000,000 tokens available.
    - Reserve pool (750,000,000)
    - Founder tokens (450,000,000)
    - Advisor tokens (150,000,000)
    - For Sale tokens (3,650,000,000)
        - 3,500,000,000 For Sale + 150,000,000 Bounty & Airdrop
- Lockups and Vesting
    - Advisor tokens vest at 30%/30%/40%, 12 weeks, 24 weeks, 365 days at close of ICO.
    - Founder tokens locked for 365 days at close of ICO.
    - Reserve tokens locked for 1095 days at close of ICO.

Explicit Vulnerability Check Results

Known Vulnerability	Results
Parity Multisig Bug 2	Not vulnerable
Callstack Depth Attack	Not vulnerable
Transaction-Ordering Dependence	Not vulnerable
Timestamp Dependency	Not vulnerable
Re-Entrancy Vulnerability	Not vulnerable
Proxy and Buffer Overflow	Not vulnerable

ISSUES DISCOVERED

Issues below are listed from most critical to least critical. Severity is determined by an assessment of the risk of exploitation or otherwise unsafe behavior.

Severity Levels

- **Informational** - No impact on the contract.
- **Low** - Minimal impact on operational ability.
- **Medium** - Affects the ability of the contract to operate.
- **High** - Affects the ability of the contract to work as designed in a significant way.
- **Critical** - Funds may be allocated incorrectly, lost or otherwise result in a significant loss.

Issues

ARAW-1 / Critical: Tokens not allocated correctly during contract initialization

Present in ArawToken.sol

Explanation

During initialization of the ArawToken contract, the reserved, founder and advisor token balances are initialized. The total supply of ArawTokens is 5,000,000,000, but only 4,550,000,000 tokens are distributed.

This leaves a remaining balance of 450,000,000 tokens that are lost.

In addition on line 39 of ArawToken the balance[founderTokens] is set based on the balances[reservedTokens] amount which results in an incorrect number of founderTokens (750,000,000 + 150,000,000 instead of only 150,000,000 founder)

Resolution

Resolved in c49dd761fc065bc62c197cb781df4583c36834df.

ARAW-2 / Informational: TimeLock.sol contains TokenTimelock contract, file name should reflect contract name

Present in TimeLock.sol

Explanation

The TokenTimelock contract is contained within the TimeLock.sol solidity file. Best practices determine that the named file should reflect the contract contained within.

Resolution

Resolved in f18e185963ad6555e21051c5154a01da061993f6.

ARAW-3 / Informational: TokenTimeLock contract not in use

Explanation

The TokenTimelock contract is not currently referenced by any other contracts. If this contract is not in use, it should be removed from the repository.

Resolution

Resolved in f18e185963ad6555e21051c5154a01da061993f6.

ARAW-4 / Critical: Possibility of tokens minted outside of token generation event

Explanation

In ArawToken.sol, the releaseTokenAdvisor function sets the balances[advisorToken] variable to be the balances[reservedTokens] amount plus the number of released tokens.

The logic for this is incorrect.

The advisorToken amount should be set to the current advisorToken amount plus the releasedTokens. The reservedTokens amount should also be decremented by the releasedTokens amount.

The above operations should also only happen if the reservedTokens amount is greater than the releasedTokens amount. Further, the Transfer event should reference the balances[reservedTokens] address rather than address(0) since new tokens are not being minted.

Resolution

Resolved in 628feabde7e145655533b64e6037fbf6839fe15a.

CONCLUSION

The reviewed smart contracts are free of security issues and well crafted.

The effort the ARAW team has put into reviewing the security of their contracts shows their commitment to security.

We look forward to seeing the success of the ARAW team and appreciate the opportunity to be a part of their story.