COINMERCENARY / SMART CONTRACT AUDIT

EinsteinCash Token

11 SEPTEMBER 2018 / 14 JULY 2019 / 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 EinsteinCash token 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 September 11, 2018 using git hash 326e124862ad8d0e6a645d93e1541ed54b03f048 the following contract files and their respective SHA256 fingerprints were reviewed:

Filename	SHA256 Fingerprint
EinsteinCash.sol	f247c5e8756369a277b5f16145a6b1c0bb491f1fecf1df96353a76cec7160753
EinsteinCash-Flattened.sol	dc9c136669a267596680c190404af199c5deb4979c12bf9596372ad0742e3229

Contract Review #2

On July 24th, 2019 using git hash 2627dbe2b7a5140ea3ae041877764ff1ff2b9e7c, the following contract files and their respective SHA256 fingerprints were reviewed:

Filename	SHA256 Fingerprint
EinsteinCash.sol	6db91b88cf5a3d22097d0ba1ab1981d5c9897edce44d69df7b07e5dfb543d413
EinsteinCash-Flattened.sol	486b72800773b5fc1d24d0cfdd2475722bd67385c73ea195ee7a5295065d97f9

AUDIT SUMMARY

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

Analysis Results

	Initial Audit	Audit #2
Design Patterns	Passed	Passed
Static Analysis	Passed	Passed
Manual Analysis	Passed	Passed
Token Allocation	Passed	Passed
Network Behavior	Passed	Passed

Test Results

- No unit test coverage available.

Token Allocation Results

- Symbol: EXE
- Decimal: 8
- Total Supply: 1,000,000,000
- Functionality: Owner, Pause and Burn

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

No issues discovered.

EINSTEINCASH AUDIT CONCLUSION

July 14th, 2019

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.

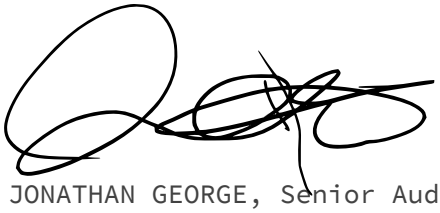
The scope of this audit was to analyze and document the EinsteinCash token contract. The audit provides practical assurance of the logic and implementation of the contracts.

CoinMercenary has reviewed the EinsteinCash smart contracts and found them to be free of security issues and logic errors.

The audit began on September 11th, 2018, with a second follow-up audit performed on July 14th, 2019. No issues were discovered during either audit.

Working with the Einstein Capital team has been a pleasure and we look forward to seeing their continued success.

Sincerely,



JONATHAN GEORGE, Senior Auditor