COINMERCENARY / SMART CONTRACT AUDIT

JSECoin Token

23 JUNE 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 JSECoin 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 June 23, 2018 using git hash e21e0a021c754c34380774b831badab49e6b3cd6 the following contract files and their respective SHA256 fingerprints were reviewed:

Filename	SHA256 Fingerprint
JSECoinCrowdsaleConfig.sol	e4e42468f8a7b4fe161961a39d53501b620bb162a7fa82300bb8739741b87873
JSEToken.sol	c77fd79df02bab6a41bcba3114216a964a9ceb2ac8747fe52bb358935d34c8f7
JSETokenSale.sol	8f0f15a5ba979228a3fa43bd0692b904e1b142870bf45444d673adb1fa7c204f
OperatorManaged.sol	ac5f165af114ae2b10ee0b9caaa73ea9b88da16ba4db85f630b7b038402f3a07

Remediation Audit

Not required.

AUDIT SUMMARY

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

Analysis Results

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

Test Results

- Basic test coverage available.

Token Allocation Results

- Symbol: JSE
- MultiSig wallet available.
- 10,000,000,000 tokens available.
- ERC223

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

JSE-1 / High: Whitelist is not checked for majority of JSE token purchases

Present in JSETokenSale.sol, lines 226 - 230.

Explanation

In the buyTokens() payable function, a comment claims “All accounts need to be whitelisted to purchase.” However, the logic in the line after that requires msg.value to be greater than or equal to CONTRIBUTION_MAX_NO_WHITELIST (15.8 ETH) before the whitelist is checked.

This allows any stakeholder sending less than 15.8 ETH to participate in the token sale without prior whitelist approval.

Resolution

Resolved via conversation with JSECoin discussing whitelisting abilities.

CONCLUSION

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

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

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