Source Code Review

Prepared for ICash Ltd. • November 2017

V1.3

1. Table of Contents

1. Table of Contents

2. Executive Summary

3. Introduction

4. Findings
    4.1. Missing “public” modifiers in SaleTracker methods
    4.2. No methods to retrieve the list of purchases in SaleTracker

5. Closing Remarks

2. Executive Summary

In November 2017, Tokensoft engaged Coinspect to perform a security audit of the ICash Network contracts on behalf of ICash Ltd. The contract was received in the following repository release:
https://github.com/icash-io/platform-contract/releases/tag/v0.2 corresponding to the commit 65d071aa69a8a8a9180a233c01a75b0de0ec8db3.

The objective of the audit was to evaluate the security of the smart contracts. During the assessment, Coinspect identified the following issues:

High-Risk	Medium Risk	Low Risk	Zero Risk
0	0	2	0

The Tokensoft team fixed all issues in the release:

https://github.com/icash-io/platform-contract/releases/tag/v0.3 corresponding to the commit d517a4de4f22e507a388a69182d32c80f6356da1

3. Introduction

The contract “SaleTracker” enable the purchase of tokens with an option payment code, as a reference. The funds received are transferred to the contract owner. The Contract “MultiSigWallet” is the standard Consensys/Gnosis multi-signature wallet smart-contract.
A whitebox security audit was conducted on these smart contracts. The contract “TestToken” is a sample ERC-20 compliant token.

The present report was completed on November 10th, 2017, by Coinspect. The report includes all issues identified in the audit.

The following checks, related to best practices, were performed:

- Confusion of the different method calling possibilities: send(), transfer(), and call.value()
- Missing error handling of external calls
- Erroneous control flow assumptions after external calls
- The use of push over pull for external calls
- Lack of enforcement of invariants with assert() and require()
- Rounding errors in integer division
- Fallback functions with higher gas limit than 2300
- Functions and state variables without explicitly visibility
- Missing pragmas to for compiler version
- Race conditions, such as contract Reentrancy
- Transaction front running
- Timestamp dependence
- Integer overflow and underflow
- Code blocks that consumes a non-constant amount of gas, that grows over block gas limit.
- Denial of Service attacks
- Suspicious code or underhanded code.

4. Findings

4.1. Missing “public” modifiers in SaleTracker methods

FIXED, Low Risk

All methods in SaleTracker are public, but they lack the “public” modifier, which is added by default. It’s a good smart-contract programming practice to clearly distinguish between private and public methods, to prevent confusion.

Recommendations

Add the “public” modifier to the methods: setEnforceAddressMatch() , purchase() and sweep().

4.2. No methods to retrieve the list of purchases in SaleTracker

FIXED, Low Risk

The SaleTracker smart-contract does not provide a method to be called off-chain to retrieve a list of purchasers. Therefore, obtaining the list must rely on observing the PurchaseMade events. Events may be unreliable in case of blockchain reorganizations. To obtain the list of purchases by using off-chain smart-contract calls, an dynamic array containing purchaser addresses must be also stored.

Recommendations

If the gas cost of storing elements in the additional list is not prohibitive for this particular use case, we recommend storing the purchasers addresses and provide the purchase listing functionality. This can increase the reliability of obtaining the purchaser list in case of any problem with the web3 event dispatching system.

5. Closing Remarks

It has been a pleasure to work with Tokensoft. The issues reported in the audit were corrected promptly. We believe the last release of the contracts are free from defects. The scope of the present security audit is limited to smart contract code. It does not cover the technologies and designs related to these smart contracts, nor the frameworks and wallets that communicate with the contracts, nor the operational security of the company that created and will deploy the audited contracts. This document should not be read as investment or legal advice.