ABDK

# TrustCoin Audit

6th April 2017

**Dmitry Khovratovich** dmitry@abdkconsulting.com

**Mikhail Vladimirov** mikhail.vladimirov@gmail.com

This is a report on the third round of audit of Trustcoin smart contracts. Most of the findings from the first two rounds are already fixed at the time of writing and thus are not included into this report. The audit was ordered in the form of quick review, and we committed to the following review plan:

- Design and code quality review;
- ERC-20 compliance review;
- Documentation review;
- Test review.

The first round of audit also dealt with security provisions and migration protocol correctness review. After we finished the first round, the Trustcoin architects followed our recommendations with removing the migration functionality. This version of review takes these changes into account.

# 1. Introduction

TrustCoin contract is an Ethereum token contract, which aims to comply with the ERC-20 standard. It also provides additional functionality, which includes a new approval function and an ability to publish an information message about migration to a new token contract. We studied the smart contract code and tests as in commit 5c1a87. The smart contract files we looked at are

| File | Description |
|---|---|
| ERC20TokenInterface.sol | Interface for ERC-20 |
| Trustcoin.sol | Contract for Trustcoin token, which implements ERC-20 and additional functionality |

# 2. Design

This section contains our findings related to Trustcoin smart contract design. Overall design looks solid simple and completely relevant to the task solved. No major design problems were identified.

## 2.1. Redundant Logging

Trustcoin smart contract contains functionality that allows contract owner to publish on blockchain an information message about migration to a new token contract. This message is saved on blockchain twice: once inside log message and another time in contract's field. This looks redundant.

## 2.2. Multiple Ways to Set Contract Owner

Contract owner is an address that is allowed to publish migration information message. This guarantees that such message, one published, is authentic and is originated from WeTrust organization. This address is passed to the contract at deployment time as constructor parameter. Also contract's owner may transfer ownership to another address via special method. So there are two ways to set owner's address: via constructor at deployment time and later via separate method call. This looks redundant. If address, that deployed the contract, would automatically become an initial owner of the smart contract with ability to transfer ownership to different address, this should be enough for all use cases, but deployment process will be simpler and more safe.

# 3. ERC-20 Compliance

Trustcoin smart contract is supposed to be ERC-20 compatible, so this sections contains our findings related to ERC-20 compatibility of Trustcoin.

## 3.1. Event Logging

ERC-20 says that `Transfer` event is "triggered when tokens are transferred". It is not clear whether transfer of zero tokens and transfer from some address to itself counts. Trustcoin logs this event in both cases. For the sake of gas economy and taking into account that `Transfer` event are usually used to track incoming/outgoing transactions for particular address, we would not recommend to log `Transfer` event in either of mentioned cases.
In contrast, regarding the `Approval` event, ERC-20 says that it is "triggered whenever `approve(address _spender, uint256 _value)` is called". This event is also logged when `compareAndApprove` function is called, which looks reasonable.

### 3.2. Meta Information

Fields `name`, `decimals` and `symbol` were once proposed to be included into ERC-20, but the community didn't agree. The idea is to use separate token registry contracts to store meta-information about token contracts.

## 4. Code Quality

Overall code quality looks good. The code is simple and readable, and have enough comments to understand it without preliminary knowledge about how it works.

## 5. Documentation Sufficiency

Documentation inside the code looks good and sufficient for those who will use Trustcoin smart contract either directly or from another smart contracts.

## 6. Test Sufficiency

This section contains our finding regarding sufficiency of tests.

### 6.1. Return Values of Non-Constant Functions Are Not Checked

Tests does not check return values of non-constant functions, while returning correct value is part of function's contract and is worth checking.

## 7. Final Recommendations

In this section we issue our recommendations to the designers.

1. Remove redundant code unless it is really necessary (see Section 2.2);
2. Do not log events when this is not explicitly required by ERC-20 and no changes in the state of the contract were actually made;
3. Find some way to check return values of non-constant functions in tests and add such checks.

## 8. Authors

**Dmitry Khovratovich** is a security and cryptography expert, an author of deanonymization attacks on Bitcoin, a designer of Equihash and Argon2 algorithms.

**Mikhail Vladimirov** is a cryptocurrency expert and a software developer with 20 years of coding experience. He is an author of the split contract TriWallet and the Approve attack on the ERC-20 standard.