Transcodium
Smart Contract: Review

Mikhail Vladimirov and Dmitry Khovratovich

18th January 2018

This document describes the issues, which were found in the Transcodium smart contract during the code review performed by ABDK Consulting.

1. Introduction

We've been asked to review the Transcodium smart contract available at commit 62a60. We were asked to assume that the OpenZeppelin contracts inherited as a library were correct.

2. Transcodium

In this section we describe issues related to the smart contract defined in the Transcodium.sol.

2.1 Documentation and Readability Issues

This section lists documentation issues, which were found in the smart contract, and the cases where the code is correct, but too involved and/or complicated to be verified or analyzed.

1. Line 8: instead of `SimpleToken` there should be `Transcodium` (used as contract name).
2. Line 9: the contract in a comment is mentioned as an example smart contract, but it looks like the real one.
3. Line 10: a clarification in the comment is required: the names of the creators should be specified.

2.2 Unclear Behavior or Flaw

This section lists issues of the smart contract, where the contract behavior is unclear: the business logic might be violated here, but the documentation and functional requirements are not sufficiently documented to make a clear decision.

Line 29: the method `issueTokens` allows the owner to transfer his tokens from the token issuer. Taking into account the fact that the owner can make any address to become token issuer, it turns out that the owner could effectively transfer tokens from any address. Is it a proper behavior?

2.3 Suboptimal Code

This section lists suboptimal code patterns, which were found in the token smart contract.

1. Line 4: the token smart contract which is mentioned is not actually used actually, so only StandardToken smart contract is being imported by it.
2. Line 32: a different event should be logged there (probably together with `Transfer` event) to make token issuing distinguishable from normal transfers.

2.4 Overflow Issues

Line 30: `SafeMath.sub` method is being used as the part of business logic, not as the last line of defence. The code would become more readable if balance check is explicit.

3. Our Recommendations

Based on our findings, we recommend the following:

1. Check the issues marked as “unclear behavior” against functional requirements.
2. Refactor the code to remove suboptimal parts.
3. Fix the documentation, readability and other (minor) issues.