MovieCoin
Token Smart Contract

Mikhail Vladimirov and Dmitry Khovratovich

21th November 2018

This document describes the audit process of the MovieCoin Token smart contract performed by ABDK Consulting.

1. Introduction

We've been asked to review the MovieCoin Token smart contract given in a private access to the BANKEX repository.

2. Moviecoin

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

2.1 Major Flaws

This section lists major flaws, which was found in the smart contract.

1. Line 379: the function `reclaimToken` contains many tokens that return false for the successful transfer. Such tokens will not be reclaimable. It might be better to change it to simple (non-safe) transfer. Moreover, there are tokens that add a fee on top of the requested balance. You would be unable to retrieve any such tokens from the balance. Consider adding token amount to reclaim.
2. Line 247: the assignment makes impossible to know what exact value was subtracted from allowance. In this case the method almost as insecure as simple approve(`address`, `uint256`) method. Open Zeppelin already fixed this in modern releases.

2.2 Documentation Issues

This section lists documentation issues, which were found in the smart contract.

Line 147: the mapping `mapping (address => mapping (address => uint256))` has two keys, and the meaning of these keys is not obvious without documentation comment.

2.3 Arithmetic Overflow Issues

This section lists issues of token smart contract related to the arithmetic overflows.

Line 37,64: assignments `assert(c / _a == _b)` and `assert(c >= _a)` are relies on undocumented overflow behavior in Solidity. It would be better to check arguments for potential overflow before adding them.

2.4 Suboptimal Code

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

1. Line 223-224: surrounding brackets are redundant in these lines.
2. Line 314: the function `_transferOwnership` is called in only one place and this looks redundant. Consider inlining in where it is called.
3. Line 393: the variable `totalSupply_` has constant value. Consider replacing with constant to save the storage space and gas.

2.5 EIP-20 Compliance

1. Line 14,130-132: the parameter names differ from the standard. For events parameter names should be a part of public API.

3 Our Recommendations

Based on our findings, we recommend the following:

1. Fix the major issue.
2. Fix arithmetic overflow issues.
3. Refactor the code to remove suboptimal parts.
4. Fix the documentation issues.