LLC “BELOV IT LAB”
Moscow, Bersenevskaya embankment 8c1
belovitlab.ru smartcontract.ru
tel.: 8 800 222 34 68 e-mail: info@smartcontract.ru

CONFIDENTIAL

EXPERT FINDINGS

based on the results of the source code audit of smart contract by **BelovItLab** company

Expert Advisory Board:

Chief Technical Officer / _____ / M. Vasilenko

General Manager / _____ / V. Belov

Content of Findings:

INTRODUCTION

1.1. General Points

The expert findings show the results of the work on security analysis of the smart contact source code (hereinafter – the System) owned by jujx.io (hereinafter – the Company) as well as recommendations to eliminate the detected weaknesses (vulnerabilities) and to increase the protection level.

1.2. Abbreviations Used

Table 1.2-1. Abbreviations Used

Abbreviation	Interpretation
IS	Information security
IT System	Information system
EVM	Ethereum virtual machine
ERC20	Accepted standard of token in Ethereum ecosystem
Ether	Crypto currency in Ethereum network
ICO	Initial coin offering

1.3. Results

"Belov IT Lab" specialists have conducted an analysis of source code protection of the System smart contract during the period from 26.10.18 to 30.10.18.

In accordance with the best practices, the Contractor specialists offered recommendations for eliminating the weaknesses. The weaknesses can be considered acceptable because they don't bear any risks for the smart contract. You can acquaint with the list of proposals below.

The smart contract logic analysis by the Contractor team has shown that the contract does not have any functions for unauthorized fund withdrawal from the contract balance. Also, no opportunities for manipulating the smart contract methods or hidden fund withdrawal methods have been detected.

2. Work Principles

2.1. IS Threats

The Company's information resources may be subject to the following IS threats: threats of confidentiality breach, integrity and availability.

A confidentiality breach threat is aimed at disclosing information which is confidential to the Company. Once the threat is carried out, the information becomes available to the people who are not supposed to have access to it – to a number of employees, customers, partners, competitors, and third parties of the Company.

An integrity breach threat is aimed at modifying or misinterpreting information thus leading to changes in its structure or meaning, its complete or partial destruction.

An availability breach threat (a service denial threat) implies a failure in accessing the information resource by the users of the information system.

The basic IS audit principle is checking the possibility of implementation of threats that affect the System information resources within the given intruder model.

2.2. Intruder Model

A person or a group of people who have or have not entered into a collusion and can potentially carry out threats to the IS, infringe the information resources of the System and prejudice the interests of the Company are considered possible violators of the System's information security of the Company.

Basic threats of confidentiality and information integrity breaches as well as the threat of the System to deny service to the Company's clients are considered threats to the IS.

A deliberately operating intruder can pursue the following causes (as well as their various combinations):

- malicious service denial;
- enhancement of own privileges;
- unauthorized altering of information crucial in terms of business.

The work involved the usage of external and internal intruder models.

2.2.1. External Intruder

The following external intruder models are used during the infiltration test:

- an external intruder from the Internet who has access to the Ethereum network and knowledge of the tested system (since the contract source code is open) but doesn't have the rights in it.

2.2.2. Internal Intruder

The following internal intruder models are used during the infiltration test:

- an internal intruder from the Ethereum network who has the knowledge of the tested system and the rights in it.

3. Smart Contract Source Code Analysis

3.1. List of Detected Weaknesses

3.1.1. High Criticality Vulnerabilities

No high criticality vulnerabilities detected

3.1.2. Average Criticality Vulnerabilities

No average criticality vulnerabilities detected

3.1.3. Low Criticality Vulnerabilities

3.1.3.1. Incomplete Implementation of ERC20 Specification

List of Functions:

- transfer From(address from, address to, uint tokens),
- approve(address spender, uint tokens),
- allowance(address tokenOwner, address spender)
- Approval(address indexed tokenOwner, address indexed spender, uint tokens) event

Recommendations:

The above-listed functions should be added

3.1.3.2. SafeMath Contract Outdated Code

Recommendations:

Update to the latest version

3.1.3.3. Solidity Version Is Not Fixed

Recommendations:

Fix the Solidity version (line 1)

3.1.3.4. Implicit Sending of Inverse Value in "buy" Function

Recommendations:

Add return (line 93)

3.1.3.5. No "onlyStronghands" Access Modifier in “exit" Function

Recommendations:

Add "onlyStronghands" access modifier to the “exit" function (line 110)

3.1.3.6. Using address(0)

Recommendations:

Use the 0x0000000000000000000000000000000000000000 address instead of address(0) (lines 254, 258, 264, 267)

3.1.3.7. Sqrt - Does Not Show Actual Square Root

Description:

It is not critical with large numbers since the error is minimal