CYBER UNIT
Cyber Security
Strategic Partner

www.cyberunit.tech

STAK Smart Contract Audit

Date: March 12, 2021
Report for: Jigstack
By: CyberUnit.Tech

This document may contain confidential information about IT systems and intellectual property of the customer as well as information about potential vulnerabilities and methods of their exploitation.
The report containing confidential information can be used internally by the customer, or it can disclose publicly after all vulnerabilities are fixed – upon the decision of the customer.

Scope and Code Revision Date

File	Jigstack.sol
SHA1 Hash	09d6590d8f0b9578c11ed0ef8fff2b250e0414af
Date	12.03.2021

Table of contents

Introduction

This report presents the findings of the security assessment of Customer` s smart contract and its code review conducted between March 10th 2021 - March 12th 2021.

Scope

The scope of the project is Jigstack smart contract, which was shared in the project folder:
Jigstack.sol
We have scanned this smart contract for commonly known and more specific vulnerabilities. Here are some of the widely known vulnerabilities that considered (the full list includes them but is not limited to them):

- Reentrancy
- Timestamp Dependence
- Gas Limit and Loops
- DoS with (Unexpected) Throw
- DoS with Block Gas Limit
- Transaction-Ordering Dependence
- Style guide violation
- Transfer forwards all gas
- ERC20 API violation
- Compiler version not fixed
- Unchecked external call - Unchecked math
- Unsafe type inference
- Implicit visibility level

Executive Summary

According to the assessment, Jigstack smart contracts security risk is very low; no issues were found for the smart contract.

Our team performed an analysis of code functionality, manual audit and automated checks with Slither and remixed IDE. All issues found during automated investigation manually reviewed and application vulnerabilities presented in the Audit overview section. A general overview presented in the AS-IS section and all encountered matters can be found in the Audit overview section.

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to tokens loss.
High	High-level vulnerabilities are difficult to exploit. However, they also have a significant impact on smart contract execution, e.g. public access to crucial functions.
Medium	Medium-level vulnerabilities are essential to fix; however, they can't lead to tokens loss.
Low	Low-level vulnerabilities are mostly related to outdated or unused code snippets.
Lowest / Code Style / Best Practice	Lowest-level vulnerabilities, code style violations and info statements can't affect smart contract execution and can generally be ignored.

AS-IS overview

IERC20

IERC20 is a standard ERC20 interface.

SafeMath

SafeMath is the standard OpenZeppelin library for math operations used to prevent integer overflows.

Context

Context is the standard library for providing sender context.

ERC20Detailed

ERC20Detailed is the standard OpenZeppelin smart contract for descriptive ERC20 functions.

ERC20

ERC20 is the standard OpenZeppelin smart contract for ERC20 token.

Jigstack

Jigstack is a smart contract for custom ERC20 token.

Jigstack has following parameters and structs:

- string private constant _name = "Jigstack";
- string private constant _symbol = "STAK";
- uint8 private constant _decimals = 18;
- address private constant _teamWallet= 0x0875f465b064156b02465fC8657b21E105D863f5;
- uint256 internal constant _tokenUnit = 10**18;
- uint256 internal constant _billion = 10**9;
- uint256 internal constant _totalSupply = 3 * _billion * _tokenUnit;

Jigstack contract has following functions:

- constructor - public function that mints total supply to team wallet

Audit overview

Critical

No critical vulnerabilities found.

High

No high vulnerabilities found.

Medium

No high vulnerabilities found.

Low

No low vulnerabilities found.

Lowest / Code style / Best Practice

No best practice issues were found.

Conclusion

Smart contracts within the scope were manually reviewed and analyzed with static analysis tools. For the contract, high-level description of functionality presented in As-is overview section of the report.

The audit report contains all found security vulnerabilities and other issues in the reviewed code.

Security engineers found no issues.

Disclaimer

The smart contracts given for audit have analyzed following the best industry practices at the date of this report, concerning: cybersecurity vulnerabilities and issues in smart contract source code, the details of which disclosed in this report, (Source Code); the Source Code compilation, deployment and functionality (performing the intended functions).

The audit doesn't make warranties on the security of the code. It also cannot be considered as a sufficient assessment regarding the utility and safety of the system, bug free status or any other statements of the contract. While we have done our best in conducting the analysis and producing this report, it is essential to note that you should not rely on this report only. We recommend proceeding with several independent audits and a public bug bounty program to ensure the security of smart contracts.

Technical Disclaimer

Smart contracts are deployed and executed on a blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. Thus, the audit can’t guarantee specific security of the audited smart contracts.