Contract Checker

SMART CONTRACT SECURITY AUDIT OF:

INDICA TOKEN (By WEEDO)

Table of Contents

SUMMARY

CONTRACTCHECKER received an application for smart contract security audit of INDICA Token on December 1, 2021, from the project team to discover if any vulnerability in the source code of the INDICA project as well as any contract dependencies. Standard tests have been performed, using Static Analysis and Manual Review techniques

The auditing process focuses to the following considerations with collaboration of an expert team

- Functionality test of the Smart Contract to determine if proper logic has been followed throughout the whole process.
- Manually detailed examination of the code line by line by experts.
- Live test by multiple clients using Testnet.
- Analysing failure preparations to check how the Smart Contract performs in case of any bugs and vulnerabilities.
- Checking whether all the libraries used in the code are on the latest version.
- Analysing the security of the on-chain data.

Project Summary

Project Name:	INDICA Token (Part of WEEDO Project)
Web Site:	https://weedoproject.app/
Platform:	Binance Smart Chain
Token Type:	BEP20
Language:	Solidity
Platforms & Tools:	Remix IDE, Truffle, Truffle Team, Ganache, Solhint, VScode, Mythril, Contract Library
Contract address:	0x3a3542c03e781a4c24f59721713799c7fa0b3d00
Link Address:	https://bscscan.com/token/0x3a3542c03e781a4c24f59721713799c7fa0b3d00

INDICA Token is released as part of WEEDO project. INDICA token allows it's investors to earn FLOKI and 2% of each transaction goes to WEEDO LP to be burned for ever.

Audit Result Summary

✓ INDICA Token has **PASSED** the smart contract audit successfully

(Other unknown security vulnerabilities are not included in the audit responsibility scope)

Audit Result: **Passed**

Ownership: **Renounced**

KYC Verification: **KYC done with CoinMooner**

Audit Date: December 1, 2021

Audit Team: CONTRACTCHECKER

Privileges of Ownership

If a privilege has high risk, it will be written in RED, in this case there are no risk.

Owner can enable trade but can't disable - Usually for Private Sale / Public Sale

Owner can transfer ownership to a new account

Owner can enable or disable the swap and liquify - Disabling will not make distribution to Marketing etc.

Owner can exclude / include from fees

Owner can exclude / include from reward

Owner can exclude / include from anti-whale

Owner can change maximum transfer amount rate for anti-whale system

Owner can change marketing and/or charity wallet address

Owner can change the min amount of token needed top swap and liquify

Owner can modify fee percentages (total limit is 20)

Owner can include any wallet to whitelist

OVERWIEW

This Audit Report mainly focuses on overall security of INDICA Smart Contract. Contractchecker team scanned the contract and assessed overall system architecture and the smart contract codebase against vulnerabilities, exploits, hacks, and back-doors to ensure its reliability and correctness.

Auditing Approach and Applied Methodologies

Coinchecker team has performed rigorous test procedures of the project

➢ Code design patterns analysis in which smart contract architecture is reviewed to ensure it is structured according to industry standards and safe use of third-party smart contracts and libraries.

➢ Line-by-line inspection of the Smart Contract to find any potential vulnerability like race conditions, transaction-ordering dependence, timestamp dependence, and denial of service attacks.

➢ Unit testing Phase, we coded/conducted custom unit tests written for each function in the contract to verify that each function works as expected.

➢ Automated Test performed with our in-house developed tools to identify vulnerabilities and security flaws of the Smart Contract.

The focus of the audit was to verify that the Smart Contract System is secure, resilient and working according to the specifications. The audit activities can be grouped in the following three categories:

Security
Identifying security related issues within each contract and the system of contract.

Sound Architecture
Evaluation of the architecture of this system through the lens of established smart contract best practices and general software best practices.

Code Correctness and Quality
A full review of the contract source code. The primary areas of focus include:
- Accuracy

- Readability
- Sections of code with high complexity
- Quantity and quality of test coverage

Risk Classification

Vulnerabilities are classified in 3 main levels as below based on possible effect to the contract.

High level vulnerability

Vulnerabilities on this level must be fixed immediately as they might lead to fund and data loss and open to manipulation.

Medium level vulnerability

Vulnerabilities on this level also important to fix as they have potential risk of future exploit and manipulation.

Low level vulnerability

Vulnerabilities on this level are minor and may not affect the smart contract execution.

Vulnerability Checklist

№	Description.	Result
1	Compiler warnings.	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log.	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed
19	Cross-function race conditions.	Passed

| 20 | Safe Zeppelin module. | Passed |
| 21 | Fallback function security. | Passed |

Manual Audit:

For this section the code was tested/read line by line by our developers. Additionally, Remix IDE's JavaScript VM and Kovan networks used to test the contract functionality.

Smart Contract SWC Attack Test

SWC attack test is not in scope of standard audit process

Automated Audit

Remix Compiler Warnings

It throws warnings by Solidity's compiler. No issues found.

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and ContractChecker and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (ContractChecker) owe no duty of care towards you or any other person, nor does ContractChecker make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and ContractChecker hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, ContractChecker hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against ContractChecker, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.