

Audits BY ICSA



www.icsa.website



Disclaimer

ICSA audits and reports should not be considered as a form of project's "advertisement" and does not cover any interaction and assessment from "project's contract" to "external contracts" such as Pancakeswap or similar.

ICSA does not provide any warranty on its released reports. We should not be used as a decision to invest into an audited project please do your own research. ICSA provides transparent reports to all its "clients" and to its "clients participants" and will not claim any guarantee of bug-free code within its Smart Contract.

Each company or project shall be liable for its own security flaws and functionalities. ICSA presence is to analyze, audit and assess the client's smart contract's code.



Scope of Work

The main focus of this report/audit, is to document an accurate assessment of the condition of the smart contract and whether it has any security flaws in the implementation of the contract.

Octo Labs team agreed and provided us with the files that needed to be tested (Through Github, EtherScan, files, etc.). **ICSA** will be focusing on contract issues and functionalities along with the projects claims from smart contract to their website, white paper and repository where available, which has been provided by the project.

Code is reviewed manually and with the use of software using industry best practices.



Project



Welcome to the world of **OCTO Labs** – an exciting realm of Meme, Utility and Rewards. Our native token, \$OCTO, is designed to revolutionize the way you think about investments and rewards. Buckle up as we take you on a thrilling journey through the innovative oceanomics of OCTO Labs!



Overview

ICSA was commissioned by **Octo Labs** to perform an audit of their smart contract:

0x75b5C2EEfd23d2354bD8133d7bD1D0ae8D21eCcb*

Blockchain → Ethereum



The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.



Contract Details

Token Name - Octo Labs

Token Description - Meme Token

Compiler Version - v0.8.20

Current Holders - 4 Address

Current Transaction Count - 4

Max Supply - 1,000,000

Token Ticker - OCTO

Decimals - 18

LP Lock - N/A

KYC'd by - ICSA*

Buy Fee - 4% (LP Prov 1%)

Sell Fee - 4% (LP Prov 1%)

Socials



[OCTO Telegram](#)



[OCTO Website](#)



[OCTO Twitter](#)



Tokenomics

Contract Owner/Deployer

0x763e88FC5b8d140B55B28a
9Ed7B00293b0b9FBa0*

Liquidity Share - 75% of taxes are added to the Liquidity Pool ensuring continued stability for the project long term.

75%

25%

Liquidity Providers - 100% of taxes go to the development wallet

Development - 25% of taxes go towards the continued development of the Octo Labs project and to help achieve road map goals.



Owner Privileges

Notes

- The owner has some privileges/authority to make SOME changes.
 - Ownership **HAS NOT** been renounced.
 - The owner can not make changes to fees.

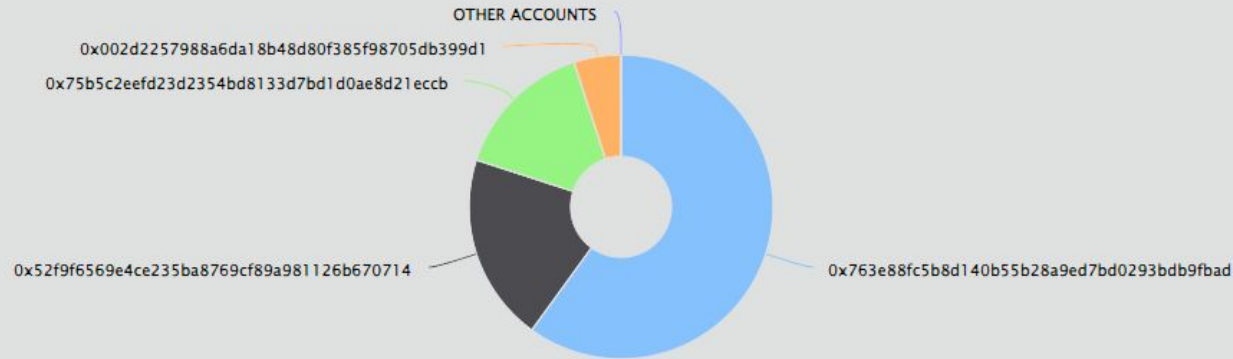




Top 100 Holders

OCTO Labs Top 100 Token Holders

Source: Etherscan.io



The total supply of 1 Million tokens are held by the top 100 holders.
The #1 wallet holds 60%% (600,000) tokens



Adjustable Functions

WRITE FUNCTIONS

1. Approve
2. Distribute Liquidity Mining Rewards
3. Get Reward
4. Release Vested Tokens
5. Renounce Ownership
6. Set Uniswap Pair
7. Stake
8. Transfer
9. Transfer From
10. Transfer Ownership
11. Vest Additional Tokens
12. Withdraw



Vulnerabilities

Passed = No Issues detected. Code is in good working order

Low Issue = Low-level weakness/vulnerabilities are mostly related to outdated, unused etc. code snippets, that can't have significant impact on execution.

High Issue = High-level weakness/vulnerabilities

SCAN RESULTS

SWC-100 → Function Default Visibility = **PASSED**

SWC-101 → Integer Overflow and Underflow = **PASSED**

SWC-102 → Outdated Compiler Version = **PASSED**

SWC-103 → Floating Pragma = **PASSED**

SWC-104 → Unlocked Call Return Value = **PASSED**



Vulnerabilities

SCAN RESULTS

SWC-105 → Unprotected Ether Withdrawal = PASSED

SWC-106 → Unprotected SELF DESTRUCT Instruction = PASSED

SWC-107 → Reentrancy = PASSED

SWC-108 → State Variable Default Visibility = PASSED

SWC-109 → Uninitialized Storage Pointer = PASSED

SWC-110 → Assert Violation = PASSED

SWC-111 → Use of Deprecated Solidity Functions = PASSED

SWC-112 → Delegatecall to Untrusted Callee = PASSED



Vulnerabilities

SCAN RESULTS

SWC-113 → DoS with Failed Call = **PASSED**

SWC-114 → Transaction Order Dependence = **PASSED**

SWC-115 → Authorization Through Tx. Origin = **PASSED**

SWC-116 → Block Values as a Value for Time = **PASSED**

SWC-117 → Signature Malleability = **PASSED**

SWC-118 → Incorrect Constructor Name = **PASSED**

SWC-119 → Shadowing State Variables = **PASSED**

SWC-120 → Weak Source of Randomness From Chain Attributes = **PASSED**



Vulnerabilities

SCAN RESULTS

SWC-121 → Missing Protection Against Signature Replay Attacks = **PASSED**

SWC-122 → Lack of Proper Signature Verification = **PASSED**

SWC-123 → Requirement Violation = **PASSED**

SWC-124 → Write to Arbitrary Storage Location = **PASSED**

SWC-125 → Incorrect Inheritance Order = **PASSED**

SWC-126 → Insufficient Gas Griefing = **PASSED**

SWC-127 → Arbitrary Jump with Function Type Variable = **PASSED**

SWC-128 → DoS with Block Gas Limit = **PASSED**



Vulnerabilities

SCAN RESULTS

SWC-129 → Typographical Error = PASSED

SWC-130 → Right-to-Left Override Control Character = PASSED

SWC-131 → Presence of Unused Variables = PASSED

SWC-132 → Unexpected Ether Balance = PASSED

SWC-133 → Hash Collisions with Multiple Variable Length Arguments = PASSED

SWC-134 → Message Call with Hardcoded Gas Amount = PASSED

SWC-135 → Code with no effects = PASSED

SWC-136 → Unencrypted Private Data On-Chain = PASSED



Scan Results

All 37 Vulnerabilities **Passed**

Please Note:

No issues found within the code!

There are no functions that can affect the security of the contract.



Manual Review

The manually read source code of **Octo Labs** has revealed no issues

NOTES

The contract is feature-rich but complex, we have thorough tested and audited to ensure security and efficiency.

Functions are generally safe due to role restrictions, but misuse of admin roles could be dangerous, team has been KYC through **ICSA**



Overall Assessment

Satisfactory!

Octo Labs has successfully passed the ICSA Audit!



June 27th, 2024



Closing Notes

Enhance the security of your crypto smart contracts with **ICSA** - the company you can trust with your digital assets. Contact us today to schedule an audit and benefit from our cutting-edge expertise in securing your blockchain projects. **ICSA**: Your gateway to safer, more secure smart contracts.

Whilst there are limitless ownable callable functions that have the potential to be dangerous,. Trust in the team would mitigate many of these risks. Please make sure you do your own research. If in doubt please contact the project team.

Always make sure to inspect all values and variables.

This includes, but is not limited to: · Ownership · Proper Ownership Renouncement (if any) · Taxes · Transaction/Wallet Limits · Token Distributions · Timelocks · Liquidity Locks · Any other owner-adjustable settings or variables.

Thank you for choosing **ICSA**