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.
Beer Ponk team agreed and provided us with the files that needed to be tested (Through Github, BscScan, 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

Beer Ponk is an innovative meme coin that combines the exciting world of Beer Pong with the dynamics of the cryptocurrency market. Their platform offers Play2Earn and Play2Burn mechanisms which are supported by a comprehensive online store and sound community.

Overview

ICSA was commissioned by Beer Ponk to perform an audit of their smart contract:

0xBc5eF2e7f93122795F966e296aA5CC82348e8C90*

Blockchain –> BNB Chain

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 - Beer Ponk

Token Ticker - BPONK

Token Description - Meme Token

Decimals - 18

Compiler Version - v0.8.19

LP Lock - N/A

Current Holders - 1 Address

KYCd by - ICSA*

Current Transaction Count - 1

Buy Fee - 5%

Max Supply - 21, 000,000,000

Sell Fee - 5%

Socials

BPONK Telegram

BPONK Website

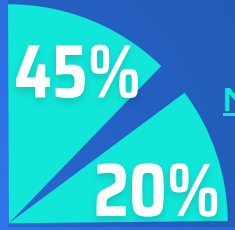
BPONK Twitter

Tokenomics

Contract Owner/Deployer

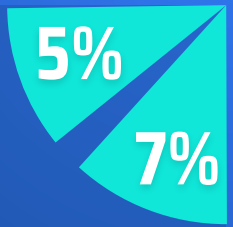
0x23Be92659e2797042f8DbA4990e1f75aF3F31542*

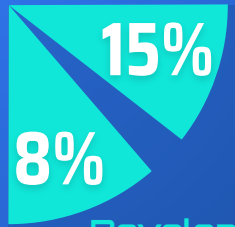
Private/Public Sale - Strategic partners and early investors will get a private sale while the rest will go as a fair public launch.



Marketing/Listing - Funding for marketing campaigns and listing on key platforms.

Earn and Burn - Promote Play2Earn and Play2Burn mechanism to reward users engagement and stabilize token value.





Liquidity - Providing liquidity to stabilize trading and promote healthy chart growth.

Team - Allocation to the team with a locked vesting period to ensure long term incentives and engagement.

Development - Supporting the tech development of the Beerponk App, online store and blockchain integration.

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.

Owner can exclude wallets from fees.

Top 100 Holders

The total supply of 21 Billion tokens are held by the top 100 holders.
The ONLY wallet holds 100% (21,000,000,000)

Adjustable Functions

WRITE FUNCTIONS

1. Approve
2. Change Pair
3. Change Threshold
4. Change Wallets
5. Enable Trading
6. Renounce Ownership
7. Rescue BNB
8. Rescue Tokens
9. Set NO Fee
10. Set Presale Address
11. Toggle Can Swap Fees
12. Transfer
13. Transfer From
14. Transfer Ownership

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!

ICSA identified some issues with the original version of the contract, this was rectified by re-deploying using our advice.

Manual Review

The manually read source code of Beer Ponk has revealed no issues

NOTES

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

This contract provides a robust implementation of an ERC-20 token with additional features for managing trading, liquidity, and fees. It includes mechanisms to protect against reentrancy and ensures that the owner has control over key parameters and functionalities.

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!

Beer Ponk has successfully passed the ICSA Audit!

June 25th, 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