



Bog Inu

Smart Contract Security Audit

Date 12/June/2021

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 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 ApeAudits and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (ApeAudits) owe no duty of care towards you or any other person, nor does ApeAudits 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 ApeAudits 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, ApeAudits hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against ApeAudits, 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.

Purpose

This document details ApeAudits' findings and recommended solutions. This audit was performed on June 12, 2021.

At the time of our audit, ownership of the contract had been renounced and 10.4 Uniswap LP tokens with an original value of 2.65 ETH + 40,933,505,459 BINU had been locked until July 11, 2021 at 23:48:49 GMT.

Token Name	Bog Inu
Token Symbol	BINU
Contract Address	0x592433C49DA39659384Fe3Ad1C6907Cd2DeE9A84
Etherscan Link	https://etherscan.io/address/0x592433c49da39659384fe3ad1c6907cd2dee9a84
Liquidity Lock Transaction	https://etherscan.io/tx/0x41662fe3e22dfb6e3a570b92936af20e895a8fa120157669459d8b736f07d142
Ownership Renounce Transaction	https://etherscan.io/tx/0x84f60397fe8d55beb2b13ddc1593de93caad4f2e89a0157709732f5ccf44559

Findings

NO.	Audit Items	Audit Subclass	Audit Subclass Result
1	Overflow Audit	N/A	Passed
2	Race Conditions Audit	N/A	Passed
3	Authority Control Audit	Permission Vulnerability Audit Excessive Auditing Authority	Passed Passed
4	Safe Design Audit	Zeppelin Module Safe Compiler Version Hard-coded Version Fallback Function Safeuse Show Coding Security Function Return Value Security Call Function Security	Passed Passed Passed Passed Passed Passed Passed
5	Denial of Service Audit	N/A	Passed
6	Gas Optimization Audit	N/A	Passed
7	Design Logic Audit	N/A	Passed

8	Malicious Event Log Audit	N/A	Passed
9	"False Deposit" Vulnerability Audit	N/A	Passed
10	Uninitialized Storage Pointers Audit	N/A	Passed
11	Arithmetic Accuracy Deviation Audit	N/A	Passed

BogInu.sol

High Severity Issues

- None.

Medium Severity Issues

- None.

Low Severity Issues

- **Issue:**
Function unlock() can be used to reclaim ownership after contract has been renounced by calling in the following order: lock() -> renounceOwnership() -> unlock().

Recommendation:
Contract ownership was renounced properly and thus this is a non-issue for the contract in its current state.

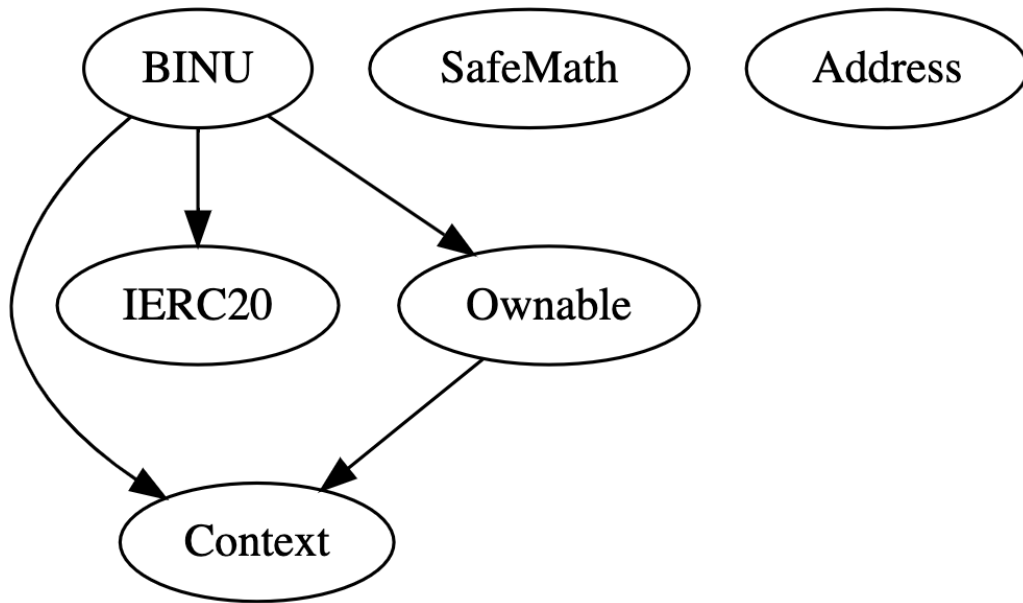
Conclusion

No issues found. Functions unlock() and renounceOwnership() have a bug that would allow ownership to be reclaimed, but it is impossible for the owner to exploit this because ownership has already been renounced.

ApeAudits note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.

Appendix B - Inheritance Graph



Appendix C - Function Information

- + Context
 - [Int] _msgSender
 - [Int] _msgData
- + [Int] IERC20
 - [Ext] totalSupply
 - [Ext] balanceOf
 - [Ext] transfer #
 - [Ext] allowance
 - [Ext] approve #
 - [Ext] transferFrom #
- + [Lib] SafeMath
 - [Int] add
 - [Int] sub
 - [Int] sub
 - [Int] mul
 - [Int] div
 - [Int] div
 - [Int] mod
 - [Int] mod
- + [Lib] Address
 - [Int] isContract
 - [Int] sendValue #
 - [Int] functionCall #
 - [Int] functionCall #
 - [Int] functionCallWithValue #
 - [Int] functionCallWithValue #
 - [Prv] _functionCallWithValue #
- + Ownable (Context)
 - [Int] <Constructor> #
 - [Pub] owner
 - [Pub] renounceOwnership #
 - modifiers: onlyOwner
 - [Pub] transferOwnership #
 - modifiers: onlyOwner
 - [Pub] geUnlockTime
 - [Pub] lock #
 - modifiers: onlyOwner
 - [Pub] unlock #
- + BINU (Context, IERC20, Ownable)
 - [Pub] <Constructor> #
 - [Pub] name
 - [Pub] symbol
 - [Pub] decimals
 - [Pub] totalSupply
 - [Pub] balanceOf
 - [Pub] transfer #
 - [Pub] allowance

- [Pub] approve #
- [Pub] transferFrom #
- [Pub] increaseAllowance #
- [Pub] decreaseAllowance #
- [Pub] isExcluded
- [Pub] totalFees
- [Pub] totalBurn
- [Pub] deliver #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Ext] excludeAccount #
- modifiers: onlyOwner
- [Ext] includeAccount #
- modifiers: onlyOwner
- [Ext] addBotToBogged #
- modifiers: onlyOwner
- [Ext] removeBotFromBogged #
- modifiers: onlyOwner
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] _transferStandard #
- [Prv] _transferToExcluded #
- [Prv] _transferFromExcluded #
- [Prv] _transferBothExcluded #
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Ext] _setTaxFee #
- modifiers: onlyOwner
- [Ext] _setBurnFee #
- modifiers: onlyOwner

(\$) = payable function

= non-constant function