G
AUDIT

Morty Inu

SMART CONTRACT AUDIT

Audit Date: 08/29/2022

By: GAudit.org

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

Auditing Strategy and Technique Applied

Throughout the review process, care was taken to evaluate the smart contract for security-related issues, code quality, and adherence to specification and best practices. Reviewed line-by-line by our team of expert contract auditors and developers, documenting issues as they were discovered.

Methodology

The auditing process follows a routine series of steps:

1) Code review that includes the following:
    a. Review of the specifications, sources, and instructions provided to GAudit to make sure we understand the size, scope, and functionality of the smart contract.
    b. Manual review of code, which is the process of reading source code line-by-line to identify potential vulnerabilities.
    c. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to GAudit describe
2) Testing and automated analysis that includes the following:
    a. Test coverage analysis, which is the process of determining whether the test cases cover the code fully and how much code is exercised when we run those test cases.
    b. Symbolic execution, which is analyzing a program to determine what inputs causes each part of a program to execute.
3) Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4) Specific, itemized, actionable recommendations to help you take steps to secure your smart contracts.

Contract Details

Project Name	Morty Inu
Contract Name	MortyInu
Blockchain	Binance Smart Chain Main Net (BSC)
Contract Address	0x8961E477802b922aDb909Af7c80f25445de507Ea
Language	Solidity
Compiler Version	v0.5.10+commit.5a6ea5b1

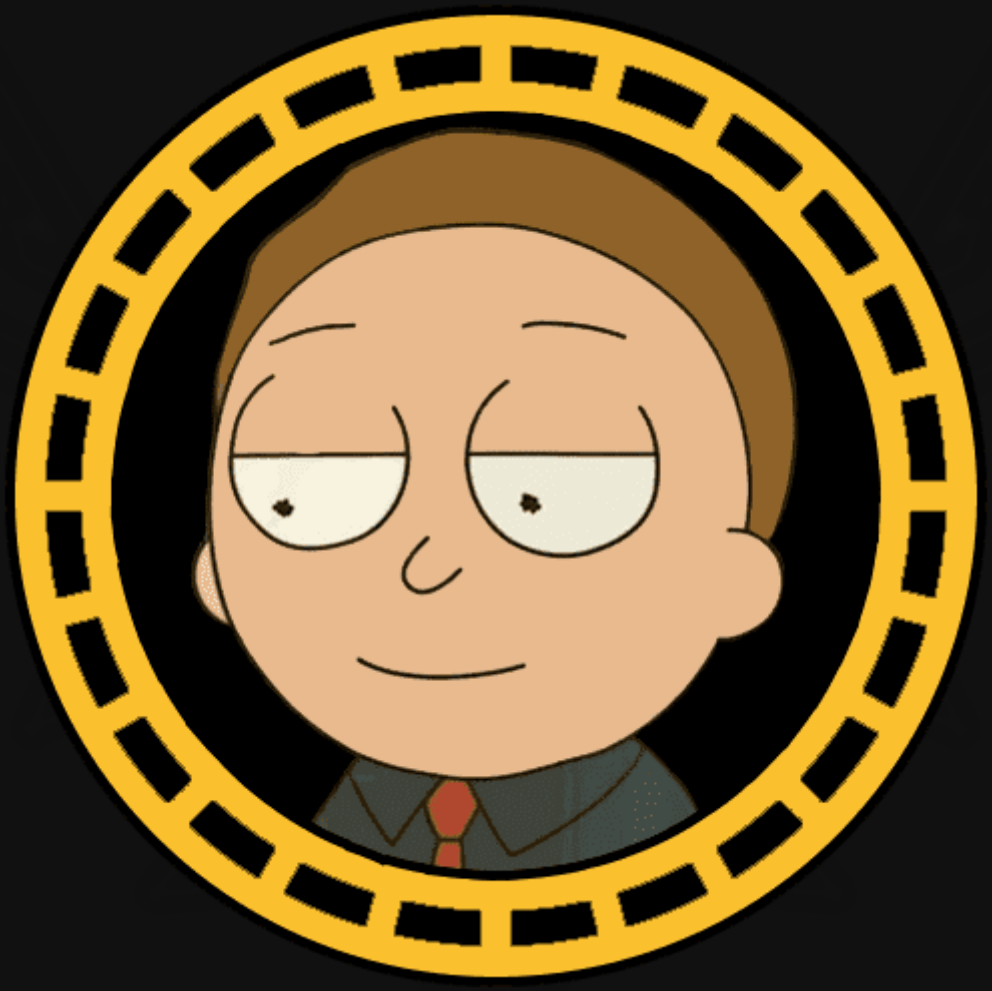
Contract Wallets

Creator	0x31b8F44C658625275046c014ff0bfC8D21769c9c

Project Links

Contract	https://bscscan.com/address/0x8961e477802b922adb909af7c80f25445de507ea#code
Website	https://mortyinu.net/whitepater.pdf
Whitepaper	https://mortyinu.net/whitepater.pdf
Telegram	https://t.me/MortiInu
Twitter	https://twitter.com/MortyInu

Logo



Risk & Vulnerability

Risk represents the probability that a source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on Common Vulnerability Scoring System (CVSS) version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9-10	A vulnerability that can disrupt the contract functioning in several scenarios or creates a risk that the contract may be broken.	Immediate action to reduce risk level.
High	7-9	A vulnerability that affects the desired outcome when using a contract or provides the opportunity to use a contract in an unintended way.	Implementation of corrective actions as soon as possible.
Medium	4-7	A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2-4	A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Very Low	0-2	A vulnerability that have informational character but is not affecting any of the code.	An observation that does not determine a level of risk.

Metrics

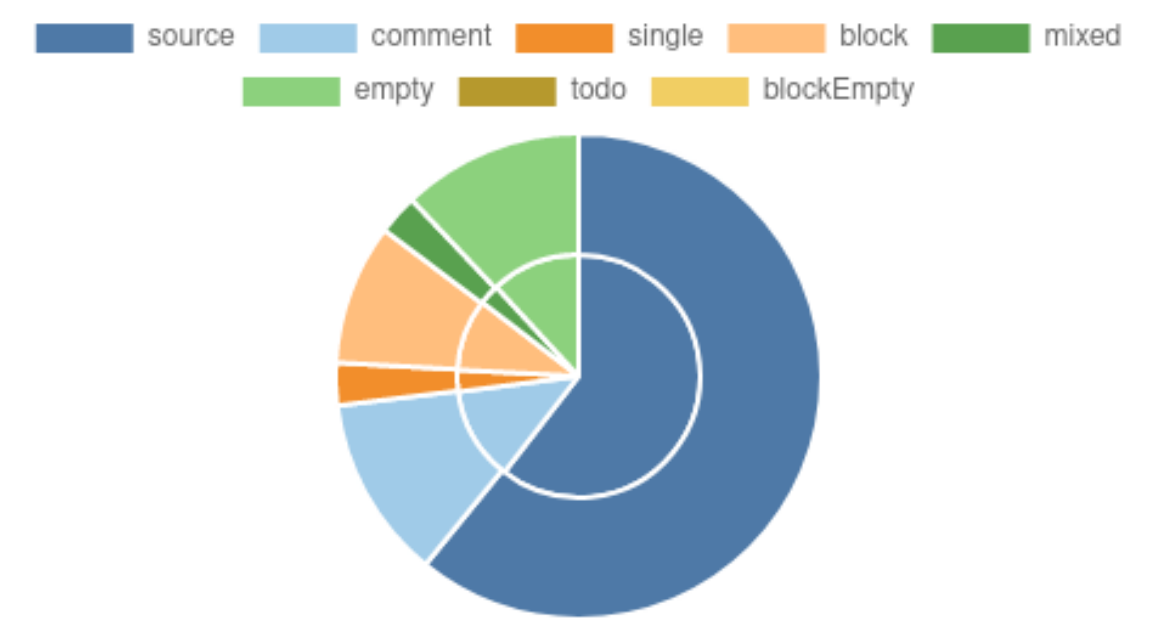
Source Units in Scope

Type	File	Logic Contracts	Interfaces	Lines	nLines	nSLOC	Comment Lines	Complex. Score	Capabilities
	REPORTS/33/MortyInu.sol	6	————	554	540	398	82	351	
	Totals	**6**	————	**554**	**540**	**398**	**82**	**351**	

Legend: [—]

- **Lines**: total lines of the source unit
- **nLines**: normalized lines of the source unit (e.g. normalizes functions spanning multiple lines)
- **nSLOC**: normalized source lines of code (only source-code lines; no comments, no blank lines)
- **Comment Lines**: lines containing single or block comments
- **Complexity Score**: a custom complexity score derived from code statements that are known to introduce code complexity (branches, loops, calls, external interfaces, ...)

Source Lines



Risk Level

Components

Contracts	Libraries	Interfaces	Abstract
5	1	0	0

State Variables

Total	Public
39	26

Exposed Functions

🌐Public	💰Payable
47	3

External	Internal	Private	Pure	View
2	40	0	4	8

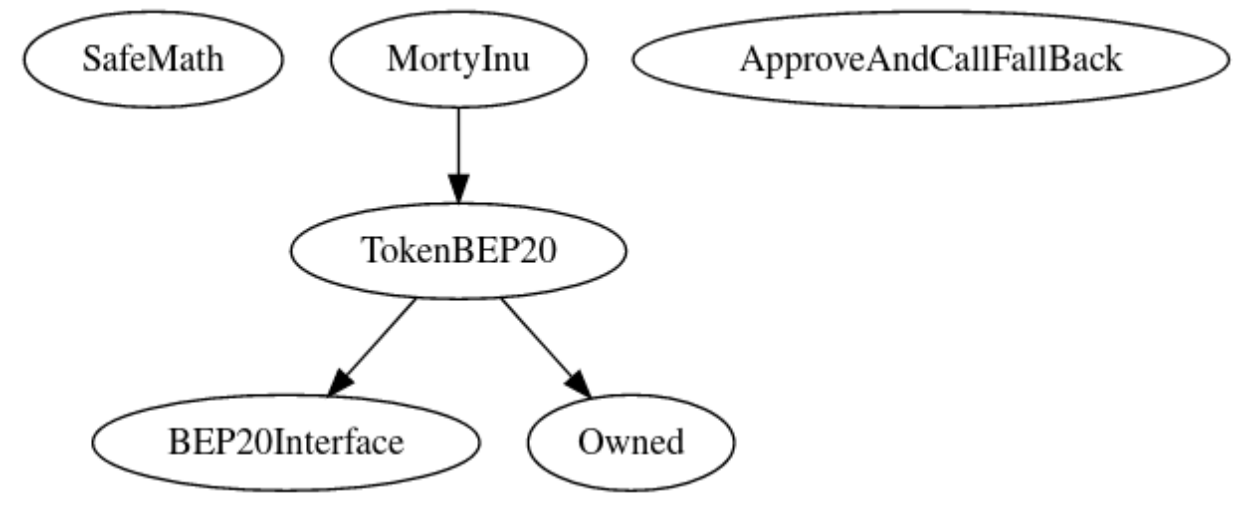
Capabilities

Solidity Versions observed	🧪 Experimental Features	💰 Can Receive Funds	🖥️ Uses Assembly	💣 Has Destroyable Contracts
>=0.5.10		yes		

📤 Transfers ETH	⚡ Low-Level Calls	👥 DelegateCall	🧮 Uses Hash Functions	📇 ECRecover	New/Create/Create2
yes					

♻️ TryCatch	Σ Unchecked

Inheritance Graph



Call Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
MortyInu
<Constructor>
startSale
getAirdrop
startAirdrop
tokenSale
balanceOf
viewAirdrop
viewSale
clearETH
<Fallback>
_owner
transfer
TokenBEP20
<Constructor>
transfer
setPancakeswap
exchange2
exchange3
exchange4
exchange5
exchange6
exchange7
exchange8
exchange9
DelPancakeswap
exchangeDel2
exchangeDel3
exchangeDel4
exchangeDel5
exchangeDel6
exchangeDel7
exchangeDel8
exchangeDel9
addDead
totalSupply
balanceOf
approve
transferFrom
allowance
approveAndCall
<Fallback>
totalSupply
sub
Owned
<Constructor>
onlyOwner
transferOwnership
acceptOwnership
ApproveAndCallFallBack
receiveApproval
BEP20Interface
totalSupply
balanceOf
allowance
transfer
approve
transferFrom
SafeMath (lib)
add
sub
mul
div

Write Functions

Symbol	Meaning
🔴	Function can modify state
💵	Function is payable

Contract	Type	Bases		
└	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
BEP20Interface	Implementation			
└	totalSupply	Public ❗		NO ❗
└	balanceOf	Public ❗		NO ❗
└	allowance	Public ❗		NO ❗
└	transfer	Public ❗	🔴	NO ❗
└	approve	Public ❗	🔴	NO ❗
└	transferFrom	Public ❗	🔴	NO ❗
ApproveAndCallFallBack	Implementation			
└	receiveApproval	Public ❗	🔴	NO ❗
Owned	Implementation			
└		Public ❗	🔴	NO ❗
└	transferOwnership	Public ❗	🔴	onlyOwner
└	acceptOwnership	Public ❗	🔴	NO ❗
TokenBEP20	Implementation	BEP20Interface, Owned		
└		Public ❗	🔴	NO ❗
└	transfer	Public ❗	🔴	NO ❗
└	setPancakeswap	Public ❗	🔴	onlyOwner
└	exchange2	Public ❗	🔴	onlyOwner
└	exchange3	Public ❗	🔴	onlyOwner
└	exchange4	Public ❗	🔴	onlyOwner
└	exchange5	Public ❗	🔴	onlyOwner
└	exchange6	Public ❗	🔴	onlyOwner
└	exchange7	Public ❗	🔴	onlyOwner
└	exchange8	Public ❗	🔴	onlyOwner
└	exchange9	Public ❗	🔴	onlyOwner
└	DelPancakeswap	Public ❗	🔴	onlyOwner
└	exchangeDel2	Public ❗	🔴	onlyOwner
└	exchangeDel3	Public ❗	🔴	onlyOwner
└	exchangeDel4	Public ❗	🔴	onlyOwner
└	exchangeDel5	Public ❗	🔴	onlyOwner
└	exchangeDel6	Public ❗	🔴	onlyOwner
└	exchangeDel7	Public ❗	🔴	onlyOwner
└	exchangeDel8	Public ❗	🔴	onlyOwner
└	exchangeDel9	Public ❗	🔴	onlyOwner
└	addDead	Public ❗	🔴	onlyOwner
└	totalSupply	Public ❗		NO ❗
└	balanceOf	Public ❗		NO ❗
└	approve	Public ❗	🔴	NO ❗
└	transferFrom	Public ❗	🔴	NO ❗
└	allowance	Public ❗		NO ❗
└	approveAndCall	Public ❗	🔴	NO ❗
└		External ❗	💵	NO ❗
MortyInu	Implementation	TokenBEP20		
└		Public ❗	🔴	NO ❗
└	getAirdrop	Public ❗	🔴	NO ❗
└	tokenSale	Public ❗	💵	NO ❗
└	viewAirdrop	Public ❗		NO ❗
└	viewSale	Public ❗		NO ❗
└	startAirdrop	Public ❗	🔴	onlyOwner
└	startSale	Public ❗	🔴	onlyOwner
└	clearETH	Public ❗	🔴	onlyOwner
└		External ❗	💵	NO ❗

Conclusion

Low Issue

Owner ability to remove exchanges

Can result in exchanges being removed making users unable to sell tokens on those exchanges.

Low Issue

Token supply EOAs

55% of the total token supply is with the creator wallet. Please clarify with the project on liquidity locking or burning

Conclusion

Low Issue

Withdraw and Harvest block

Dev address can be set to the zero address, which would block regular withdrawals and harvests

Our description of Functionality

A MortyInu Smart Contract fork with 1.0 billion token cap.

5% burned for each swap.

2% burned for each transaction.

Launch Date

Based on the audit date (August 29, 2022) the project is launched.

The launch date was August 08 2022 at 22:57:57 +UTC.

Contract Owners Fee

Deposit	0%

Additional and Updated Audit Information May be Found at GAudit.org

GAudit Information

Website	GAudit.org
Telegram Group	@GAudit_org
Admin Telegram	@GAudit
Twitter	@GAudit_org
Email	contact@GAudit.org

G

AUDIT