TOKEN AUDIT

Bingus Network

Smart Contract Security Audit

Audit Report

Sep, 2021

Table of contents

Project Introduction	2
Auditing Methodologies applied:	3
Static Analysis	4
Auditing Tools	4
Tokenomics	5
Issues Checking	6
Severity Issue Categories	8
Issues Found	9
Automated Testing	13
Functional View	16
Inheritance Chart	17
Audit Findings Results	19

Project Introduction

Bingus is a hairless sphynx cat that took the internet by storm with his adorable cuteness. He captured the imagination of many and quickly became a meme.

Name	Bingus Network
Total Supply	100 million
Type	BSC Token
Website	https://bingus.io/
Platform	Binance Smart Chain
Apeswap Supply	3,140,000
Holders	
Deployed Contract	0x12AdaDddC8d86081561a3ff107A2Cb347779e717

Token Audit Team performed a security audit for Bingus Network smart contracts during the period of Sep 24, 2021 to Sep 25, 2021.

Auditing Methodologies applied:

- In this audit, we can review the code listed below.
- The overall quality of code.
- Whether the implementation of BEP 20 standards.
- Whether the code is secure.
- Gas Optimization
- Code is safe from reentrancy and other vulnerabilities

Manual Audit

- Manually analyzing the source code line-by-line in an attempt to identify security vulnerabilities.
- Gas Consumption and optimization
- Assessing the overall project structure, complexity & quality.
- Checking whether all the libraries used in the code of the latest version.

Automated Audit

- Projects can be Automated using these tools with Slither, Manticore, Sol Graph others.
- Performing Unit testing.
- Symbolic execution, which is analyzing a program to determine what inputs cause each part of a program to execute.

➔ 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.

Static Analysis

Static Analysis of Smart Contracts was done to identify contract vulnerabilities. In this step a series of automated tools are used to test security of smart contracts.

Auditing Tools

Language: Solidity

Platform and tools: Slither, Manti-Core, VScode, Solhint, Solc-select, Solidity-coverage

Audit Aim: The focus of the audit was to verify whether the smart contract is secure, resilient, and working according to the standard specs. The audit activity can be categories into three types

- Security
- Sound Architecture
- Code Correctness and Quality

Tokenomics:

Distribution:

With the launch of Bingus Network, we have offered holders of Bingus 2.0 the opportunity to claim the new token based on their previous holdings. Therefore, we have the distribution available to those holders, we additionally reserve some $BINGUS for any disputes

Of the 100,000,000 BINGUS supply:

74,043,708 74.0 % is made available for Bingus 2.0 holders to claim.

20,707,403 20.7 % is held in reserve for any disputes

3,140,000 3.1 % is paired with the BNB removed at snapshot in ApeSwap Liquidity

2,108,889 2.1% is held by the project for promotions, competitions, CEX/DEX listings etc.

Unclaimed allocation or unused disputes shall be returned to the project. The use of which shall be decided by the community (burn, staking, CEX, DEX etc.)

Please find the below whitepaper:

https://bingus.io/documents/Bingus-Network-Whitepaper.pdf

Issues Checking

We have scanned this smart contract code for commonly known and more specific Vulnerabilities that are below listed:

SN	Issue Description	Status
1	Re-entrancy	Verified
2	Compiler errors	Verified
3	Timestamp Dependence	Verified
4	Unsafe external calls	Verified
5	Gas Limit and Loops	Verified
6	DoS with Block Gas Limit	Verified
7	Private user data leaks	Verified
8	Code clones, functionality duplication	Verified
9	Style guide violation	Verified
10	Costly Loop	Verified
11	Balance equality	Verified
12	Unchecked math	Verified

13	Integer overflow/underflow	Verified
14	Cross-function Race Condition	Verified
15	Fallback function security	Verified
16	Data Consistency	Verified
17	Balance equality	Verified
18	ERC20 API violation	Verified
19	Deployment Consistency	Verified
20	Arithmetic accuracy	Verified
21	Transaction-Ordering Dependence (TOD) / Front Running	Verified
22	Address hardcoded	Verified
23	Scoping and Declarations	Verified
24	Implicit visibility level	Verified
25	Call Depth Attack (deprecated)	Verified

Severity Issue Categories

Every issue in this report was assigned a severity level from the following:

Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to token loss etc.
High	The issue affects the ability of the contract to compile or operate in a significant way.
Medium	Issues on this level could potentially bring problems and should eventually be fixed.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.
Informational	The issue has no impact on the contract's ability to operate.

Issues Found

- Critical severity issues - No Critical severity issues found.
- High severity issues - No Critical severity issues found.
- Medium severity issues - No medium severity issues found.
- Low severity issues - 0 low severity issues were found.
- Informational - 4 Informational severity issues were found.

High	Medium	Low	Informational
0	0	0	4

Informational level severity issues

1. Incorrect version of solidity

   Severity: Informational

   pragma solidity ^0.8.2;

   Description:

   Contracts should be deployed using the same compiler version/flags with which they have been tested. Locking the pragma (for e.g., by not using ^ in pragma solidity 0.8.2) ensures that contracts do not accidentally get deployed using an older compiler version with unfixed bugs..

```
Pragma version^0.8.2 (bingus.sol#16) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity
```

   Suggestion:

   Lock the pragma version.

2. Conformance-To-Solidity-Naming-Conventions

   Severity: Informational

   Description:

   In the contract, many function names were found to be starting with capital letters. Functions other than constructors should use mixed Case

```
Variable Bingus._TAX_FEE (bingus.sol#490) is not in mixedCase
Variable Bingus._BURN_FEE (bingus.sol#491) is not in mixedCase
Variable Bingus._CHARITY_FEE (bingus.sol#492) is not in mixedCase
Variable Bingus.ORIG_TAX_FEE (bingus.sol#496) is not in mixedCase
Variable Bingus.ORIG_BURN_FEE (bingus.sol#497) is not in mixedCase
Variable Bingus.ORIG_CHARITY_FEE (bingus.sol#498) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions
```

   Rule exceptions:

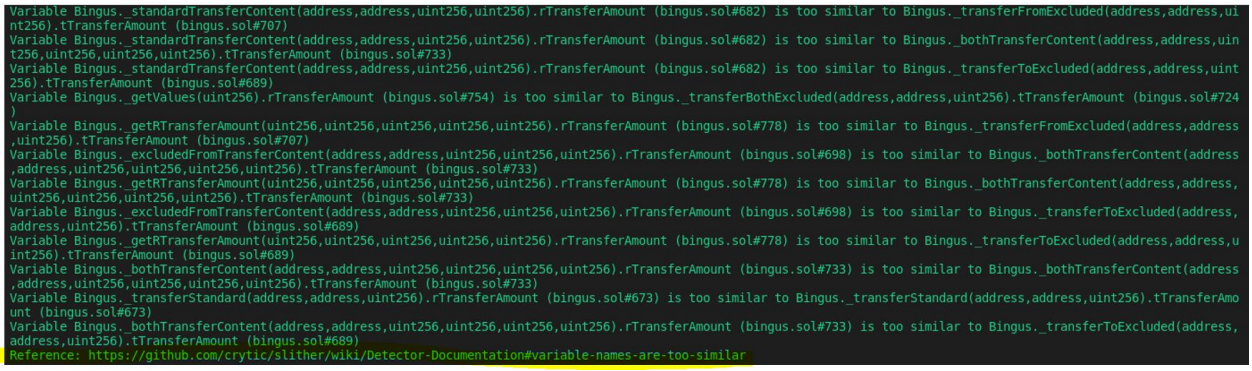
   Allow constant variable name/symbol/decimals to be lowercase (ERC20).
   Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

Suggestion:

Follow the Solidity: https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

3. Variables names are too similar

Severity: Informational



Description:

Please fine the below URL

https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

4. Public function that could be declared external

Severity: Informational

Description:

The following public functions that are never called by the contract should be declared external to save gas

```
name() should be declared external:
        - Bingus.name() (bingus.sol#505-507)
symbol() should be declared external:
        - Bingus.symbol() (bingus.sol#509-511)
decimals() should be declared external:
        - Bingus.decimals() (bingus.sol#513-515)
totalSupply() should be declared external:
        - Bingus.totalSupply() (bingus.sol#517-519)
balanceOf(address) should be declared external:
        - Bingus.balanceOf(address) (bingus.sol#521-524)
transfer(address,uint256) should be declared external:
        - Bingus.transfer(address,uint256) (bingus.sol#526-529)
allowance(address,address) should be declared external:
        - Bingus.allowance(address,address) (bingus.sol#531-533)
approve(address,uint256) should be declared external:
        - Bingus.approve(address,uint256) (bingus.sol#535-538)
transferFrom(address,address,uint256) should be declared external:
        - Bingus.transferFrom(address,address,uint256) (bingus.sol#540-544)
increaseAllowance(address,uint256) should be declared external:
        - Bingus.increaseAllowance(address,uint256) (bingus.sol#546-549)
decreaseAllowance(address,uint256) should be declared external:
        - Bingus.decreaseAllowance(address,uint256) (bingus.sol#551-554)
isExcluded(address) should be declared external:
        - Bingus.isExcluded(address) (bingus.sol#556-558)
isCharity(address) should be declared external:
        - Bingus.isCharity(address) (bingus.sol#560-562)
totalFees() should be declared external:
        - Bingus.totalFees() (bingus.sol#564-566)
totalBurn() should be declared external:
        - Bingus.totalBurn() (bingus.sol#568-570)
totalCharity() should be declared external:
        - Bingus.totalCharity() (bingus.sol#572-574)
deliver(uint256) should be declared external:
        - Bingus.deliver(uint256) (bingus.sol#576-583)
reflectionFromToken(uint256,bool) should be declared external:
        - Bingus.reflectionFromToken(uint256,bool) (bingus.sol#585-594)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
```

Suggestion:

Use the external attribute for functions never called from the contract.

Automated Testing

We have to use Automated testing Slither. It is an Automated Analysis Tool in Smart Contract.

```
Bingus.allowance(address,address).owner (bingus.sol#531) shadows:
        - Ownable.owner() (bingus.sol#427-429) (function)
Bingus._approve(address,address,uint256).owner (bingus.sol#632) shadows:
        - Ownable.owner() (bingus.sol#427-429) (function)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#local-variable-shadowing

Address.isContract(address) (bingus.sol#281-290) uses assembly
        - INLINE ASM (bingus.sol#288)
Address._functionCallWithValue(address,bytes,uint256,string) (bingus.sol#374-395) uses assembly
        - INLINE ASM (bingus.sol#387-390)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#assembly-usage

Address._functionCallWithValue(address,bytes,uint256,string) (bingus.sol#374-395) is never used and should be removed
Address.functionCall(address,bytes) (bingus.sol#334-336) is never used and should be removed
Address.functionCall(address,bytes,string) (bingus.sol#344-346) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256) (bingus.sol#359-361) is never used and should be removed
Address.functionCallWithValue(address,bytes,uint256,string) (bingus.sol#369-372) is never used and should be removed
Address.isContract(address) (bingus.sol#281-290) is never used and should be removed
Address.sendValue(address,uint256) (bingus.sol#308-314) is never used and should be removed
Bingus._getMaxTxAmount() (bingus.sol#830-832) is never used and should be removed
Bingus._getTaxFee() (bingus.sol#826-828) is never used and should be removed
Context._msgData() (bingus.sol#24-27) is never used and should be removed
SafeMath.mod(uint256,uint256) (bingus.sol#238-240) is never used and should be removed
SafeMath.mod(uint256,uint256,string) (bingus.sol#254-257) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code
```

```
Bingus._rTotal (bingus.sol#484) is set pre-construction with a non-constant function or state variable:
        - (_MAX - (_MAX % _tTotal))
Bingus.ORIG_TAX_FEE (bingus.sol#496) is set pre-construction with a non-constant function or state variable:
        - _TAX_FEE
Bingus.ORIG_BURN_FEE (bingus.sol#497) is set pre-construction with a non-constant function or state variable:
        - _BURN_FEE
Bingus.ORIG_CHARITY_FEE (bingus.sol#498) is set pre-construction with a non-constant function or state variable:
        - _CHARITY_FEE
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.2 (bingus.sol#16) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6
solc-0.8.2 is not recommended for deployment
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#incorrect-versions-of-solidity

Low level call in Address.sendValue(address,uint256) (bingus.sol#308-314):
        - (success) = recipient.call{value: amount}() (bingus.sol#312)
Low level call in Address._functionCallWithValue(address,bytes,uint256,string) (bingus.sol#374-395):
        - (success,returndata) = target.call{value: weiValue}(data) (bingus.sol#378)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls
```

```
Variable Bingus._TAX_FEE (bingus.sol#490) is not in mixedCase
Variable Bingus._BURN_FEE (bingus.sol#491) is not in mixedCase
Variable Bingus._CHARITY_FEE (bingus.sol#492) is not in mixedCase
Variable Bingus.ORIG_TAX_FEE (bingus.sol#496) is not in mixedCase
Variable Bingus.ORIG_BURN_FEE (bingus.sol#497) is not in mixedCase
Variable Bingus.ORIG_CHARITY_FEE (bingus.sol#498) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (bingus.sol#25)" inContext (bingus.sol#19-28)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements
```

```
Variable Bingus._standardTransferContent(address,address,uint256,uint256).rTransferAmount (bingus.sol#682) is too similar to Bingus._transferFromExcluded(address,address,ui
nt256).tTransferAmount (bingus.sol#707)
Variable Bingus._standardTransferContent(address,address,uint256,uint256).rTransferAmount (bingus.sol#682) is too similar to Bingus._bothTransferContent(address,address,uin
t256,uint256,uint256,uint256).tTransferAmount (bingus.sol#733)
Variable Bingus._standardTransferContent(address,address,uint256,uint256).rTransferAmount (bingus.sol#682) is too similar to Bingus._transferToExcluded(address,address,uint
256).tTransferAmount (bingus.sol#689)
Variable Bingus._getValues(uint256).rTransferAmount (bingus.sol#754) is too similar to Bingus._transferBothExcluded(address,address,uint256).tTransferAmount (bingus.sol#724
)
Variable Bingus._getRTransferAmount(uint256,uint256,uint256,uint256,uint256).rTransferAmount (bingus.sol#778) is too similar to Bingus._transferFromExcluded(address,address
,uint256).tTransferAmount (bingus.sol#707)
Variable Bingus._excludedFromTransferContent(address,address,uint256,uint256,uint256).rTransferAmount (bingus.sol#698) is too similar to Bingus._bothTransferContent(address
,address,uint256,uint256,uint256,uint256).tTransferAmount (bingus.sol#733)
Variable Bingus._getRTransferAmount(uint256,uint256,uint256,uint256,uint256).rTransferAmount (bingus.sol#778) is too similar to Bingus._bothTransferContent(address,address,
uint256,uint256,uint256,uint256).tTransferAmount (bingus.sol#733)
Variable Bingus._excludedFromTransferContent(address,address,uint256,uint256,uint256).rTransferAmount (bingus.sol#698) is too similar to Bingus._transferToExcluded(address,
address,uint256).tTransferAmount (bingus.sol#689)
Variable Bingus._getRTransferAmount(uint256,uint256,uint256,uint256,uint256).rTransferAmount (bingus.sol#778) is too similar to Bingus._transferToExcluded(address,address,u
int256).tTransferAmount (bingus.sol#689)
Variable Bingus._bothTransferContent(address,address,uint256,uint256,uint256,uint256).rTransferAmount (bingus.sol#733) is too similar to Bingus._bothTransferContent(address
,address,uint256,uint256,uint256,uint256).tTransferAmount (bingus.sol#733)
Variable Bingus._transferStandard(address,address,uint256).rTransferAmount (bingus.sol#673) is too similar to Bingus._transferStandard(address,address,uint256).tTransferAmo
unt (bingus.sol#673)
Variable Bingus._bothTransferContent(address,address,uint256,uint256,uint256,uint256).rTransferAmount (bingus.sol#733) is too similar to Bingus._transferToExcluded(address,
address,uint256).tTransferAmount (bingus.sol#689)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-are-too-similar

Bingus.slitherConstructorVariables() (bingus.sol#462-835) uses literals with too many digits:
        - _tTotal = 1000000000000 * _DECIMALFACTOR (bingus.sol#483)
Bingus.slitherConstructorConstantVariables() (bingus.sol#462-835) uses literals with too many digits:
        - _MAX_TX_SIZE = 10000000000 * _DECIMALFACTOR (bingus.sol#493)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#too-many-digits
```

```
name() should be declared external:
        - Bingus.name() (bingus.sol#505-507)
symbol() should be declared external:
        - Bingus.symbol() (bingus.sol#509-511)
decimals() should be declared external:
        - Bingus.decimals() (bingus.sol#513-515)
totalSupply() should be declared external:
        - Bingus.totalSupply() (bingus.sol#517-519)
balanceOf(address) should be declared external:
        - Bingus.balanceOf(address) (bingus.sol#521-524)
transfer(address,uint256) should be declared external:
        - Bingus.transfer(address,uint256) (bingus.sol#526-529)
allowance(address,address) should be declared external:
        - Bingus.allowance(address,address) (bingus.sol#531-533)
approve(address,uint256) should be declared external:
        - Bingus.approve(address,uint256) (bingus.sol#535-538)
transferFrom(address,address,uint256) should be declared external:
        - Bingus.transferFrom(address,address,uint256) (bingus.sol#540-544)
increaseAllowance(address,uint256) should be declared external:
        - Bingus.increaseAllowance(address,uint256) (bingus.sol#546-549)
decreaseAllowance(address,uint256) should be declared external:
        - Bingus.decreaseAllowance(address,uint256) (bingus.sol#551-554)
isExcluded(address) should be declared external:
        - Bingus.isExcluded(address) (bingus.sol#556-558)
isCharity(address) should be declared external:
        - Bingus.isCharity(address) (bingus.sol#560-562)
totalFees() should be declared external:
        - Bingus.totalFees() (bingus.sol#564-566)
totalBurn() should be declared external:
        - Bingus.totalBurn() (bingus.sol#568-570)
totalCharity() should be declared external:
        - Bingus.totalCharity() (bingus.sol#572-574)
deliver(uint256) should be declared external:
        - Bingus.deliver(uint256) (bingus.sol#576-583)
reflectionFromToken(uint256,bool) should be declared external:
        - Bingus.reflectionFromToken(uint256,bool) (bingus.sol#585-594)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#public-function-that-could-be-declared-external
./bingus.sol analyzed (6 contracts with 75 detectors), 130 result(s) found
```

Sol-hint Tool:

A linter for Solidity that provides both Security and Style Guide validations.

Coding style issues influence code readability and, in some cases, may lead to bugs in future. Smart Contracts have a naming convention, indentation and code layout issues. It's recommended to use Solidity Style Guide to fix all the issues. Consider following the Solidity guidelines on formatting the code and commenting for all the files. It can improve the overall code quality and readability

```
bingus.sol
   25:2  error  Line length must be no more than 120 but current length is 132  max-line-length
  306:2  error  Line length must be no more than 120 but current length is 160  max-line-length
  325:2  error  Line length must be no more than 120 but current length is 156  max-line-length
  344:2  error  Line length must be no more than 120 but current length is 122  max-line-length
  369:2  error  Line length must be no more than 120 but current length is 146  max-line-length
  374:2  error  Line length must be no more than 120 but current length is 149  max-line-length
  542:2  error  Line length must be no more than 120 but current length is 130  max-line-length
  552:2  error  Line length must be no more than 120 but current length is 138  max-line-length
  626:2  error  Line length must be no more than 120 but current length is 125  max-line-length
  673:2  error  Line length must be no more than 120 but current length is 159  max-line-length
  682:2  error  Line length must be no more than 120 but current length is 124  max-line-length
  689:2  error  Line length must be no more than 120 but current length is 159  max-line-length
  698:2  error  Line length must be no more than 120 but current length is 153  max-line-length
  707:2  error  Line length must be no more than 120 but current length is 159  max-line-length
  716:2  error  Line length must be no more than 120 but current length is 143  max-line-length
  724:2  error  Line length must be no more than 120 but current length is 159  max-line-length
  733:2  error  Line length must be no more than 120 but current length is 162  max-line-length
  740:2  error  Line length must be no more than 120 but current length is 128  max-line-length
  749:2  error  Line length must be no more than 120 but current length is 127  max-line-length
  758:2  error  Line length must be no more than 120 but current length is 145  max-line-length
  765:2  error  Line length must be no more than 120 but current length is 128  max-line-length
  775:2  error  Line length must be no more than 120 but current length is 150  max-line-length

✖ 22 problems (22 errors, 0 warnings)
```

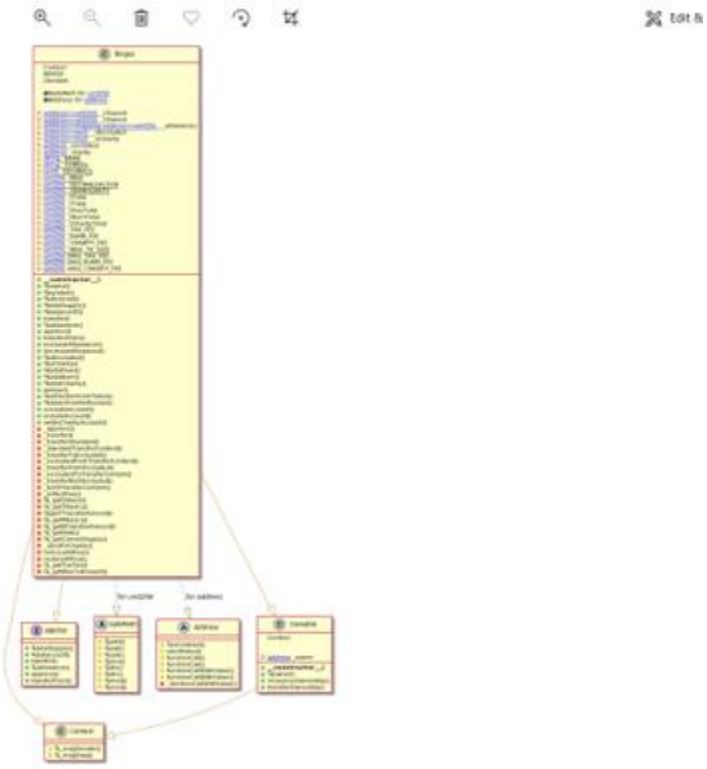
Functional View

Function	Return Type	Test Result
name	Public	Verified
symbol	Public	Verified
decimals	Public	Verified
totalSupply	Public	Verified
balanceOf	Public	Verified
transfer	Public	Verified
allowance	Public	Verified
approve	Public	Verified
TransferFrom	Public	Verified
increaseAllowance	Public	Verified
decreaseAllowance	Public	Verified
isExcludedFromReward	Public	Verified
totalFees	Public	Verified
deliver	Public	Verified
reflectionFromToken	Public	Verified

tokenFromReflection	Public	Verified
excludeFromReward	Public	Verified
includeInReward	Public	Verified
transferBothExcluded	Public	Verified

Inheritance Chart:

UML:



Audit Findings Results

There were 4 Informational found during the audit. All the mentioned findings may have an effect only in case of specific conditions performed by the contract owner. None of the critical issues were resolved.

Generally, the contracts are well written and structured. The findings during the audit have some impact on contract performance or security

Disclaimer

This audit does not provide a security or correctness guarantee of audited smart contracts You agree that your access and/or use, including but not limited to any services, products, platforms, content, will be at your Own risk. Smart contract remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond the programming language aspects that could present security risks. A report does not indicate the endorsement of any particular project or team, nor guarantee its security.


http://tokenaudit.net/


https://t.me/TokenAuditt


https://twitter.com/AuditToken