



Smart Contract Audit

FOR

MARIO BSC

DATED : 03 October 23'



AUDIT SUMMARY

Project name – MARIO BSC

Date: 03 October 2023

Scope of Audit- Audit Ace was consulted to conduct the smart contract audit of the solidity source codes.

Audit Status: **Passed**

Issues Found

Status	Critical	High	Medium	Low	Suggestion
Open	0	0	0	0	0
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1- Manual Review:

A line by line code review has been performed by audit ace team.

2- BSC Test Network: All tests were conducted on the BSC Test network, and each test has a corresponding transaction attached to it. These tests can be found in the "Functional Tests" section of the report.

3- Slither :

The code has undergone static analysis using Slither.

Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:

<https://testnet.bscscan.com/token/0x764c3AD40958E09E5381E34BEa9FB2674a579552>



Token Information

Token Address :

0x2cF1AcF496e24a774d737a316315eE63bf371b86

Name : MARIO BSC

Symbol : MARIO

Decimals: 9

Owner: 0xc280E12AaFB1C05AA0E3BAb93208FE994c701B2D

Deployer:

0xc280E12AaFB1C05AA0E3BAb93208FE994c701B2D

Token Supply: 420,000,000,000

Checksum:

388560a874c128a9b7669f396b80ede50a44d6ce

Testnet version:

The tests were performed using the contract deployed on the BSC Testnet, which can be found at the following address:

<https://testnet.bscscan.com/token/0x764c3AD40958E09E5381E34BEa9FB2674a579552>



TOKEN OVERVIEW

buy fee: 10%

Sell fee: 10%

transfer fee: 10%

Fee Privilege: Static fees

Ownership: Owned

Minting: None

Max Tx: No

Blacklist: No

Other Privileges:

- Initial distribution of the tokens



AUDIT METHODOLOGY

The auditing process will follow a routine as special considerations by Auditace:

- Review of the specifications, sources, and instructions provided to Auditace to make sure the contract logic meets the intentions of the client without exposing the user's funds to risk.
 - Manual review of the entire codebase by our experts, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
 - Specification comparison is the process of checking whether the code does what the specifications, sources, and instructions provided to Auditace describe.
 - Test coverage analysis determines whether the test cases are covering the code and how much code is exercised when we run the test cases.
 - Symbolic execution is analysing a program to determine what inputs cause each part of a program to execute.
 - Reviewing the codebase to improve maintainability, security, and control based on the established industry and academic practices.
-

VULNERABILITY CHECKLIST

- | | |
|------------------------------------|-------------------------------|
| ✓ Return values of low-level calls | ✓ Gasless Send |
| ✓ Private modifier | ✓ Using block.timestamp |
| ✓ Multiple Sends | ✓ Re-entrancy |
| ✓ Using Suicide | ✓ Tautology or contradiction |
| ✓ Gas Limitand Loops | ✓ Timestamp Dependence |
| ✓ Address hardcoded | ✓ Revert/require functions |
| ✓ Exception Disorder | ✓ Use of tx.origin |
| ✓ Using inline assembly | ✓ Integer overflow/underflow |
| ✓ Divide before multiply | ✓ Dangerous strict equalities |
| ✓ Missing Zero Address Validation | ✓ Using SHA3 |
| ✓ Compiler version not fixed | ✓ Using throw |
-

CLASSIFICATION OF RISK

Severity

Description

◆ Critical

These vulnerabilities could be exploited easily and can lead to asset loss, data loss, asset, or data manipulation. They should be fixed right away.

◆ High-Risk

A vulnerability that affects the desired outcome when using a contract, or provides the opportunity to use a contract in an unintended way.

◆ Medium-Risk

A vulnerability that could affect the desired outcome of executing the contract in a specific scenario.

◆ Low-Risk

A vulnerability that does not have a significant impact on possible scenarios for the use of the contract and is probably subjective.

◆ Gas Optimization /Suggestion

A vulnerability that has an informational character but is not affecting any of the code.

Findings

Severity

Found

◆ Critical

0

◆ High-Risk

0

◆ Medium-Risk

0

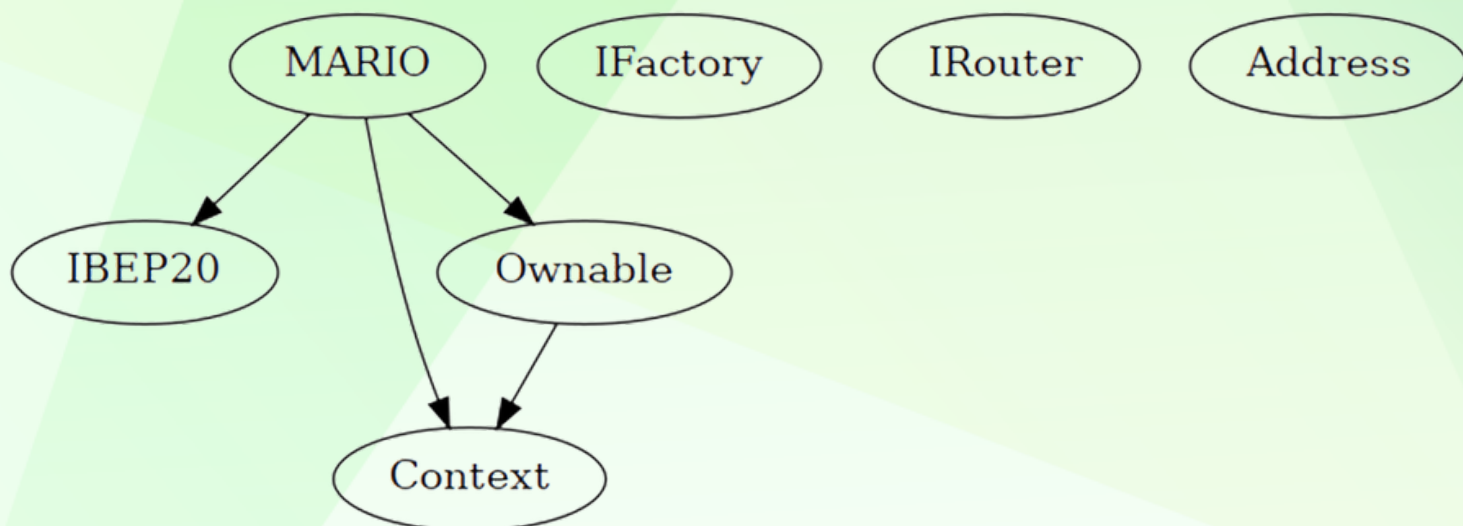
◆ Low-Risk

0

◆ Gas Optimization / Suggestions

0

INHERITANCE TREE





POINTS TO NOTE

- Owner is not able to change buy taxes (10% static)
 - Owner is not able to change sell taxes (10% static)
 - Owner is not able to change transfer taxes (10% static)
 - Owner is not able to set max wallet/transfer/buy/sell amounts
 - Owner is not able to blacklist an arbitrary wallet
 - Owner is not able to disable trades
 - Owner is not able to mint new tokens
-



STATIC ANALYSIS

```
Reentrancy in CATCEO.transferFrom(address,address,uint256) (contracts/Token.sol#251-266):
  External calls:
    - _transfer(sender,recipient,amount) (contracts/Token.sol#256)
      - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
      - router.swapExactTokensForETHSupportingFeeOnTransferTokens(tokenAmount,0,path,address(this),block.timestamp) (contracts/Token.sol#536-542)
      - address(marketingWallet).sendValue(deltaBalance) (contracts/Token.sol#523)
  External calls sending eth:
    - _transfer(sender,recipient,amount) (contracts/Token.sol#256)
      - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
  Event emitted after the call(s):
    - Approval(owner,spender,amount) (contracts/Token.sol#461)
      - approve(sender,msgSender(),currentAllowance - amount) (contracts/Token.sol#263)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#reentrancy-vulnerabilities-3

CATCEO.includeInReward(address) (contracts/Token.sol#341-352) has costly operations inside a loop:
  - excluded.pop() (contracts/Token.sol#348)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#costly-operations-inside-a-loop

Context._msgData() (contracts/Token.sol#45-48) is never used and should be removed
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#dead-code

CATCEO._rTotal (contracts/Token.sol#151) is set pre-construction with a non-constant function or state variable:
  - (MAX - (MAX % tTotal))
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#function-initializing-state

Pragma version^0.8.17 (contracts/Token.sol#6) necessitates a version too recent to be trusted. Consider deploying with 0.6.12/0.7.6/0.8.16
solc-0.8.18 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) (contracts/Token.sol#117-128):
  - (success) = recipient.call{value: amount}() (contracts/Token.sol#123)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IRouter.WETH() (contracts/Token.sol#93) is not in mixedCase
Struct CATCEO.valuesFromGetValues (contracts/Token.sol#175-183) is not in CapWords
Constant CATCEO._decimals (contracts/Token.sol#147) is not in UPPER_CASE_WITH_UNDERSCORES
Constant CATCEO._name (contracts/Token.sol#158) is not in UPPER_CASE_WITH_UNDERSCORES
Constant CATCEO._symbol (contracts/Token.sol#159) is not in UPPER_CASE_WITH_UNDERSCORES
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Redundant expression "this (contracts/Token.sol#46)" inContext (contracts/Token.sol#40-49)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#redundant-statements

CATCEO._tTotal (contracts/Token.sol#150) should be constant
CATCEO.deadWallet (contracts/Token.sol#155) should be constant
CATCEO.marketingWallet (contracts/Token.sol#156) should be constant
CATCEO.swapTokensAtAmount (contracts/Token.sol#153) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

CATCEO.pair (contracts/Token.sol#145) should be immutable
CATCEO.router (contracts/Token.sol#144) should be immutable
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-immutable
```

Result => A static analysis of contract's source code has been performed using slither,

No major issues were found in the output



CONTRACT ASSESMENT

Contract	Type	Bases			
:-----: :-----: :-----: :-----: :-----:					
⌞	**Function Name** **Visibility** **Mutability** **Modifiers**				
	IBEP20 Interface				
⌞	totalSupply	External	!		NO !
⌞	balanceOf	External	!		NO !
⌞	transfer	External	!	●	NO !
⌞	allowance	External	!		NO !
⌞	approve	External	!	●	NO !
⌞	transferFrom	External	!	●	NO !
	Context Implementation				
⌞	_msgSender	Internal	🔒		
⌞	_msgData	Internal	🔒		
	Ownable Implementation Context				
⌞	<Constructor>	Public	!	●	NO !
⌞	owner	Public	!		NO !
⌞	renounceOwnership	Public	!	●	onlyOwner
⌞	_setOwner	Private	🔒	●	
	IFactory Interface				
⌞	createPair	External	!	●	NO !
	IRouter Interface				
⌞	factory	External	!		NO !
⌞	WETH	External	!		NO !
⌞	addLiquidityETH	External	!	💰	NO !
⌞	swapExactTokensForETHSupportingFeeOnTransferTokens	External	!	●	NO !
	Address Library				
⌞	sendValue	Internal	🔒	●	



CONTRACT ASSESMENT

|||||

| ****MARIO**** | Implementation | Context, IBEP20, Ownable |||

| | <Constructor> | Public ! | | NO ! |

| | name | Public ! | | NO ! |

| | symbol | Public ! | | NO ! |

| | decimals | Public ! | | NO ! |

| | totalSupply | Public ! | | NO ! |

| | balanceOf | Public ! | | NO ! |

| | allowance | Public ! | | NO ! |

| | approve | Public ! | | NO ! |

| | transferFrom | Public ! | | NO ! |

| | increaseAllowance | Public ! | | NO ! |

| | decreaseAllowance | Public ! | | NO ! |

| | transfer | Public ! | | NO ! |

| | isExcludedFromReward | Public ! | | NO ! |

| | reflectionFromToken | Public ! | | NO ! |

| | tokenFromReflection | Public ! | | NO ! |

| | excludeFromReward | Public ! | | onlyOwner |

| | includeInReward | External ! | | onlyOwner |

| | excludeFromFee | Public ! | | onlyOwner |

| | includeInFee | Public ! | | onlyOwner |

| | isExcludedFromFee | Public ! | | NO ! |

| | _reflectRfi | Private | | |

| | _takeMarketing | Private | | |

| | _getValues | Private | | |

| | _getTValues | Private | | |

| | _getRValues | Private | | |

| | _getRate | Private | | |

| | _getCurrentSupply | Private | | |

| | _approve | Private | | |

| | _transfer | Private | | |

| | _tokenTransfer | Private | | |

| | swapAndLiquify | Private | | lockTheSwap |

| | swapTokensForBNB | Private | | |


| | bulkExcludeFee | External ! | | onlyOwner |

| | <Receive Ether> | External ! | | NO ! |



CONTRACT ASSESMENT

Legend

Symbol Meaning	
:-----: -----	
●	Function can modify state
	Function is payable



FUNCTIONAL TESTING

1- Adding liquidity (**passed**):

<https://testnet.bscscan.com/tx/0x5bb468de0ef4abce0753e30f414762dc90b36ea63c68fac6dfd544a19fd33e29>

2- Buying when excluded (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x8c641095d412048dfaff37e333d136f15a9b39c256e68779e0506dc9d012c247>

3- Selling when excluded (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xe90c2ffc042ea39b1e6e618863c19965130e148d7458de55e11df61df0e43c0f>

4- Transferring when excluded (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x5e11e49e7019b4955d9c38d752f1db8d8447057d76b6aba7752c0dcce6a4595b>

5- Buying when not excluded (10% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x91a7a42abbb1a06b6a24b98dab9eb94c29a41e30ab10bd5b27ec0e9d560818>

6- Selling when not excluded (10% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xfecd374e8b160108ff20bff22c45e5ed5caeacc07485b7dc1c42625c568a7504>

7- Transferring when not excluded (10% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x723d0e4de1e6dd533a49528e1b3c85eab549d1b2aa7f5540d8ca2cc3bcaa1420>

8- Internal swap (**passed**):

marketing wallet received ETH

<https://testnet.bscscan.com/tx/0xfecd374e8b160108ff20bff22c45e5ed5caeacc07485b7dc1c42625c568a7504>



DISCLAIMER

All the content provided in this document is for general information only and should not be used as financial advice or a reason to buy any investment. Team provides no guarantees against the sale of team tokens or the removal of liquidity by the project audited in this document. Always Do your own research and protect yourselves from being scammed. The Auditace team has audited this project for general information and only expresses their opinion based on similar projects and checks from popular diagnostic tools. Under no circumstances did Auditace receive a payment to manipulate those results or change the awarding badge that we will be adding in our website. Always Do your own research and protect yourselves from scams. This document should not be presented as a reason to buy or not buy any particular token. The Auditace team disclaims any liability for the resulting losses.



ABOUT AUDITACE

We specialize in providing thorough and reliable audits for Web3 projects. With a team of experienced professionals, we use cutting-edge technology and rigorous methodologies to evaluate the security and integrity of blockchain systems. We are committed to helping our clients ensure the safety and transparency of their digital assets and transactions.



<https://auditace.tech/>



https://t.me/Audit_Ace



https://twitter.com/auditace_



<https://github.com/Audit-Ace>
