auditace.tech

AuditAce
FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

AIDODE3.0

DATED : 4 July 23'

AUDIT SUMMARY

Project name – AIDODE3.0

Date: 4 July, 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	1	0	2
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/0xB4a99296989e6cF637c973b30bA181503BB95420

Token Information

Token Name : AIDODE3.0

Token Symbol: AIDODE3.0

Decimals: 9

Token Supply: 2,100,000,000,000

Token Address:
0x015ac3c069909F6202340B768aD17E5C38A3c349

Checksum:
666e75b39d29acabb5178c89e9848d7b41227d93

Owner:
0xd7CD8b3442A19ae5E00493605C430a6182614E35
(at time of writing the audit)

Deployer:
0xd7CD8b3442A19ae5E00493605C430a6182614E35

TOKEN OVERVIEW

Fees:

Buy Fees: 0-25%

Sell Fees: 0-25%

Transfer Fees: 0-25%

Fees Privilege: Owner

Ownership: owned

Minting: none

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: - Initial distribution of the tokens

- modifying fees

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

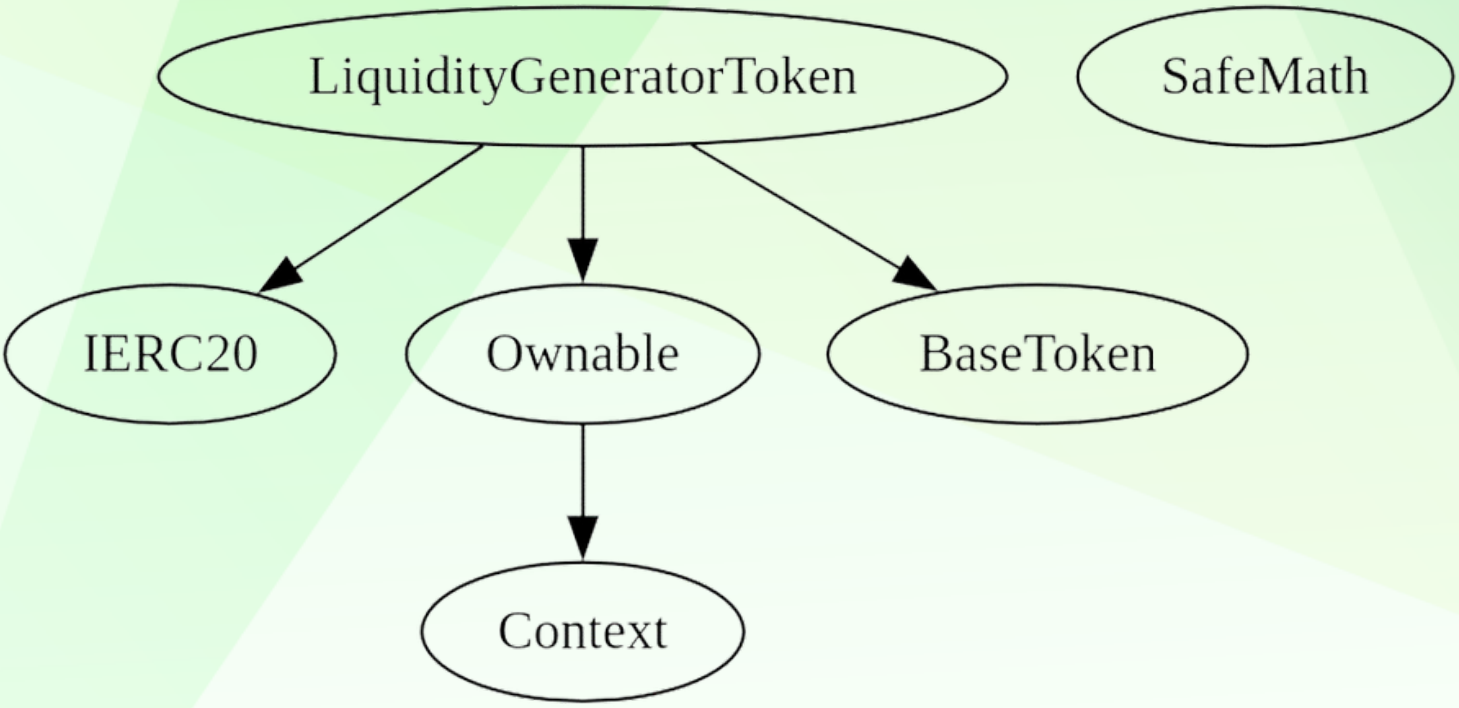
CLASSIFICATION OF RISK

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	1
Low-Risk	0
Gas Optimization / Suggestions	2

INHERITANCE TREE



POINTS TO NOTE

- Owner is able to change buy/sell/transfer fees in range 0-25%
- Owner is not able to blacklist an arbitrary address.
- Owner is not able to disable trades
- Owner is not able to set max buy/sell/transfer/hold amount to 0
- Owner is not able to mint new tokens

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
IERC20	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
└	transferOwnership	Public ❗	🛑	onlyOwner
└	_setOwner	Private 🔐	🛑	
SafeMath	Library			
└	tryAdd	Internal 🔒		
└	trySub	Internal 🔒		
└	tryMul	Internal 🔒		
└	tryDiv	Internal 🔒		
└	tryMod	Internal 🔒		
└	add	Internal 🔒		
└	sub	Internal 🔒		
└	mul	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
└	sub	Internal 🔒		
└	div	Internal 🔒		
└	mod	Internal 🔒		
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	

CONTRACT ASSESMENT

└	functionCallWithValue	Internal 🔒	🛑	
└	functionStaticCall	Internal 🔒		
└	functionStaticCall	Internal 🔒		
└	functionDelegateCall	Internal 🔒	🛑	
└	functionDelegateCall	Internal 🔒	🛑	
└	verifyCallResult	Internal 🔒		
IUniswapV2Router01	Interface			
└	factory	External ❗	|NO ❗	
└	WETH	External ❗	|NO ❗	
└	addLiquidity	External ❗	🛑 |NO ❗	
└	addLiquidityETH	External ❗	💵 |NO ❗	
└	removeLiquidity	External ❗	🛑 |NO ❗	
└	removeLiquidityETH	External ❗	🛑 |NO ❗	
└	removeLiquidityWithPermit	External ❗	🛑 |NO ❗	
└	removeLiquidityETHWithPermit	External ❗	🛑 |NO ❗	
└	swapExactTokensForTokens	External ❗	🛑 |NO ❗	
└	swapTokensForExactTokens	External ❗	🛑 |NO ❗	
└	swapExactETHForTokens	External ❗	💵 |NO ❗	
└	swapTokensForExactETH	External ❗	🛑 |NO ❗	
└	swapExactTokensForETH	External ❗	🛑 |NO ❗	
└	swapETHForExactTokens	External ❗	💵 |NO ❗	
└	quote	External ❗	|NO ❗	
└	getAmountOut	External ❗	|NO ❗	
└	getAmountIn	External ❗	|NO ❗	
└	getAmountsOut	External ❗	|NO ❗	
└	getAmountsIn	External ❗	|NO ❗	
IUniswapV2Router02	Interface	IUniswapV2Router01		
└	removeLiquidityETHSupportingFeeOnTransferTokens	External ❗	🛑 |NO ❗	
└	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External ❗	🛑 |NO ❗	
└	swapExactTokensForTokensSupportingFeeOnTransferTokens	External ❗	🛑 |NO ❗	
└	swapExactETHForTokensSupportingFeeOnTransferTokens	External ❗	💵 |NO ❗	
└	swapExactTokensForETHSupportingFeeOnTransferTokens	External ❗	🛑 |NO ❗	
IUniswapV2Factory	Interface			
└	feeTo	External ❗	|NO ❗	
└	feeToSetter	External ❗	|NO ❗	
└	getPair	External ❗	|NO ❗	
└	allPairs	External ❗	|NO ❗	
└	allPairsLength	External ❗	|NO ❗	
└	createPair	External ❗	🛑 |NO ❗	

CONTRACT ASSESMENT

└	setFeeTo	External ❗	🛑	NO ❗
└	setFeeToSetter	External ❗	🛑	NO ❗
BaseToken	Implementation			
LiquidityGeneratorToken	Implementation	IERC20, Ownable, BaseToken		
└	<Constructor>	Public ❗	💵	NO ❗
└	name	Public ❗		NO ❗
└	symbol	Public ❗		NO ❗
└	decimals	Public ❗		NO ❗
└	totalSupply	Public ❗		NO ❗
└	balanceOf	Public ❗		NO ❗
└	transfer	Public ❗	🛑	NO ❗
└	allowance	Public ❗		NO ❗
└	approve	Public ❗	🛑	NO ❗
└	transferFrom	Public ❗	🛑	NO ❗
└	increaseAllowance	Public ❗	🛑	NO ❗
└	decreaseAllowance	Public ❗	🛑	NO ❗
└	isExcludedFromReward	Public ❗		NO ❗
└	totalFees	Public ❗		NO ❗
└	deliver	Public ❗	🛑	NO ❗
└	reflectionFromToken	Public ❗		NO ❗
└	tokenFromReflection	Public ❗		NO ❗
└	excludeFromReward	Public ❗	🛑	onlyOwner
└	includeInReward	External ❗	🛑	onlyOwner
└	_transferBothExcluded	Private 🔐	🛑	
└	excludeFromFee	Public ❗	🛑	onlyOwner
└	setTaxFeePercent	External ❗	🛑	onlyOwner
└	setLiquidityFeePercent	External ❗	🛑	onlyOwner
└	setCharityFeePercent	External ❗	🛑	onlyOwner
└	setSwapBackSettings	External ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💵	NO ❗
└	_reflectFee	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_takeLiquidity	Private 🔐	🛑	
└	_takeCharityFee	Private 🔐	🛑	
└	calculateTaxFee	Private 🔐		
└	calculateLiquidityFee	Private 🔐		

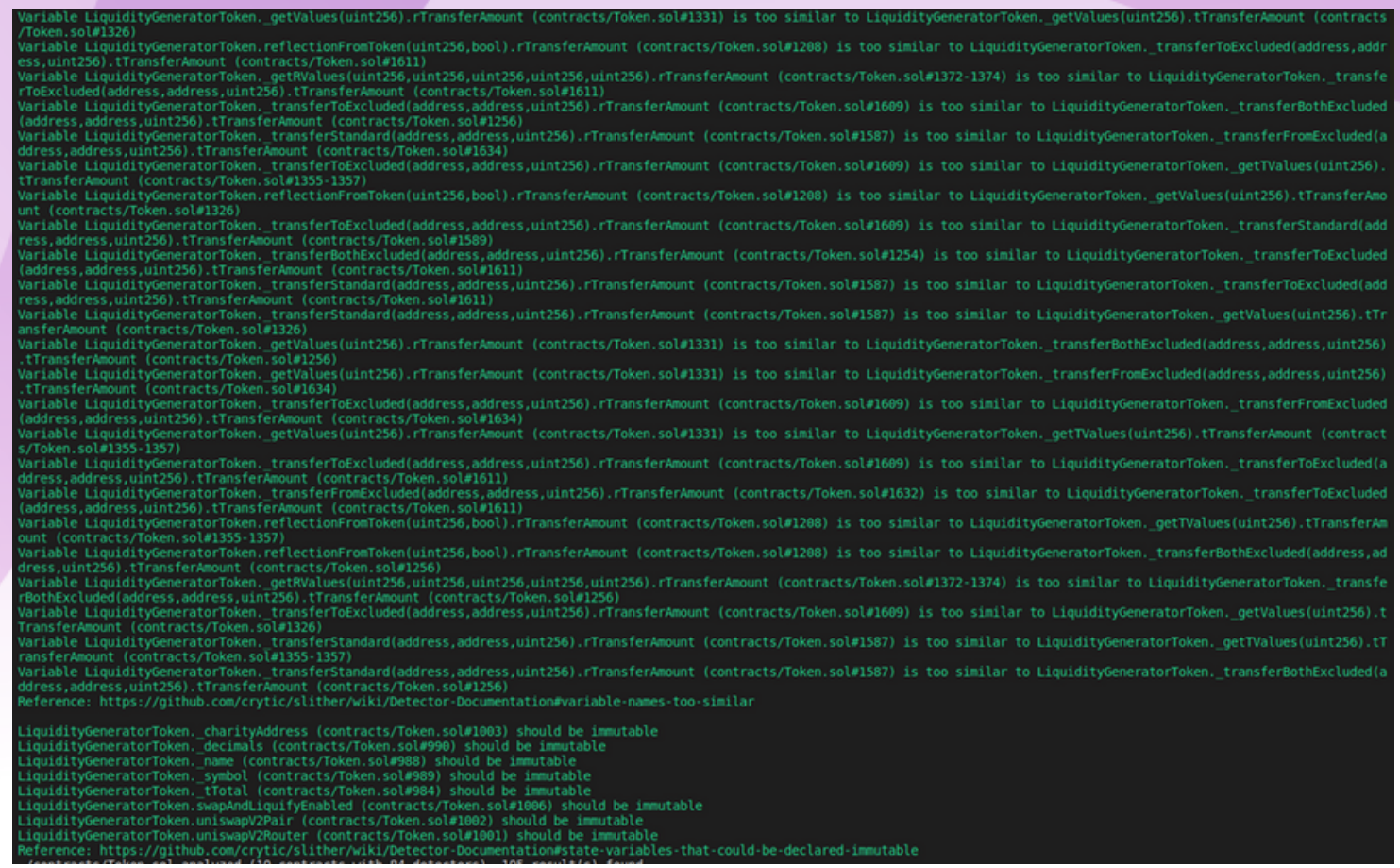
CONTRACT ASSESMENT

└	calculateCharityFee	Private 🔐		
└	removeAllFee	Private 🔐	🛑	
└	restoreAllFee	Private 🔐	🛑	
└	isExcludedFromFee	Public ❗		NO ❗
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	lockTheSwap
└	swapTokensForEth	Private 🔐	🛑	
└	addLiquidity	Private 🔐	🛑	
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	

Legend

Symbol	Meaning
🛑	Function can modify state
💵	Function is payable

STATIC ANALYSIS



```
Variable LiquidityGeneratorToken._getValues(uint256).rTransferAmount (contracts/Token.sol#1331) is too similar to LiquidityGeneratorToken._getValues(uint256).tTransferAmount (contracts
/Token.sol#1326)
Variable LiquidityGeneratorToken.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#1208) is too similar to LiquidityGeneratorToken._transferToExcluded(address,addr
ess,uint256).tTransferAmount (contracts/Token.sol#1611)
Variable LiquidityGeneratorToken._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1372-1374) is too similar to LiquidityGeneratorToken._transfe
rToExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1611)
Variable LiquidityGeneratorToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._transferBothExcluded
(address,address,uint256).tTransferAmount (contracts/Token.sol#1256)
Variable LiquidityGeneratorToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1587) is too similar to LiquidityGeneratorToken._transferFromExcluded(a
ddress,address,uint256).tTransferAmount (contracts/Token.sol#1634)
Variable LiquidityGeneratorToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._getTValues(uint256).
tTransferAmount (contracts/Token.sol#1355-1357)
Variable LiquidityGeneratorToken.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#1208) is too similar to LiquidityGeneratorToken._getValues(uint256).tTransferAmo
unt (contracts/Token.sol#1326)
Variable LiquidityGeneratorToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._transferStandard(add
ress,address,uint256).tTransferAmount (contracts/Token.sol#1589)
Variable LiquidityGeneratorToken._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1254) is too similar to LiquidityGeneratorToken._transferToExcluded
(address,address,uint256).tTransferAmount (contracts/Token.sol#1611)
Variable LiquidityGeneratorToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1587) is too similar to LiquidityGeneratorToken._transferToExcluded(add
ress,address,uint256).tTransferAmount (contracts/Token.sol#1611)
Variable LiquidityGeneratorToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1587) is too similar to LiquidityGeneratorToken._getValues(uint256).tTr
ansferAmount (contracts/Token.sol#1326)
Variable LiquidityGeneratorToken._getValues(uint256).rTransferAmount (contracts/Token.sol#1331) is too similar to LiquidityGeneratorToken._transferBothExcluded(address,address,uint256)
.tTransferAmount (contracts/Token.sol#1256)
Variable LiquidityGeneratorToken._getValues(uint256).rTransferAmount (contracts/Token.sol#1331) is too similar to LiquidityGeneratorToken._transferFromExcluded(address,address,uint256)
.tTransferAmount (contracts/Token.sol#1634)
Variable LiquidityGeneratorToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._transferFromExcluded
(address,address,uint256).tTransferAmount (contracts/Token.sol#1634)
Variable LiquidityGeneratorToken._getValues(uint256).rTransferAmount (contracts/Token.sol#1331) is too similar to LiquidityGeneratorToken._getTValues(uint256).tTransferAmount (contract
s/Token.sol#1355-1357)
Variable LiquidityGeneratorToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._transferToExcluded(a
ddress,address,uint256).tTransferAmount (contracts/Token.sol#1611)
Variable LiquidityGeneratorToken._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1632) is too similar to LiquidityGeneratorToken._transferToExcluded
(address,address,uint256).tTransferAmount (contracts/Token.sol#1611)
Variable LiquidityGeneratorToken.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#1208) is too similar to LiquidityGeneratorToken._getTValues(uint256).tTransferAm
ount (contracts/Token.sol#1355-1357)
Variable LiquidityGeneratorToken.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#1208) is too similar to LiquidityGeneratorToken._transferBothExcluded(address,ad
dress,uint256).tTransferAmount (contracts/Token.sol#1256)
Variable LiquidityGeneratorToken._getRValues(uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#1372-1374) is too similar to LiquidityGeneratorToken._transfe
rBothExcluded(address,address,uint256).tTransferAmount (contracts/Token.sol#1256)
Variable LiquidityGeneratorToken._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#1609) is too similar to LiquidityGeneratorToken._getValues(uint256).t
TransferAmount (contracts/Token.sol#1326)
Variable LiquidityGeneratorToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1587) is too similar to LiquidityGeneratorToken._getTValues(uint256).tT
ransferAmount (contracts/Token.sol#1355-1357)
Variable LiquidityGeneratorToken._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#1587) is too similar to LiquidityGeneratorToken._transferBothExcluded(a
ddress,address,uint256).tTransferAmount (contracts/Token.sol#1256)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

LiquidityGeneratorToken._charityAddress (contracts/Token.sol#1003) should be immutable
LiquidityGeneratorToken._decimals (contracts/Token.sol#990) should be immutable
LiquidityGeneratorToken._name (contracts/Token.sol#988) should be immutable
LiquidityGeneratorToken._symbol (contracts/Token.sol#989) should be immutable
LiquidityGeneratorToken._tTotal (contracts/Token.sol#984) should be immutable
LiquidityGeneratorToken.swapAndLiquifyEnabled (contracts/Token.sol#1006) should be immutable
LiquidityGeneratorToken.uniswapV2Pair (contracts/Token.sol#1002) should be immutable
LiquidityGeneratorToken.uniswapV2Router (contracts/Token.sol#1001) 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

FUNCTIONAL TESTING

Router (PCS V2):
0xD99D1c33F9fC3444f8101754aBC46c52416550D1

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0x4ea163ceef3787da794809efa0f24d060fd6226c7cb5b0f96caf86382ca597d5

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x14fc557ffe1131d34c1a1006adc597d3cff0512ba2a7b1e506a63fb02e54b2bc

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x616ee3cf42f855ea42484e9ea6d79fb26943dbbbaf8905239739739 8f47f3b2b

4- Transferring when excluded from fees (0% tax) (passed):
https://testnet.bscscan.com/tx/0x9baaac178539e6c7f299e5bee2b0b8b06b33c94ac750765de650d1e4ea776286

5- Buying from a regular wallet (0-25% tax) (passed):
https://testnet.bscscan.com/tx/0x3a9e8b8bdf36ee2b26615d73d94e656bd7c49cafdd13455335c3d7c4ba24dab9

6- Selling from a regular wallet (0-25% tax) (passed):
https://testnet.bscscan.com/tx/0xf11c669f1b5e8294b0ddadf2dd1d0ff433aba9dde606b5c5bd799d44062d9ffa

FUNCTIONAL TESTING

7- Transferring from a regular wallet (0-25% tax) (passed):

https://testnet.bscscan.com/tx/0xa1d7e8e1adf8307454e52b8853b7862b016d46c59d3165ec1d136ce7e399a468

8- Internal swap (marketing bnb + auto-liquidity) (passed):

https://testnet.bscscan.com/tx/0xf11c669f1b5e8294b0ddadf2dd1d0ff433aba9dde606b5c5bd799d44062d9ffa

FUNCTIONAL TESTING

Category: Centralization
Subject: Fee setting and updating
Severity: Medium
Status: not applicable
Overview:
The contract allows the owner to set and update various fees, including tax, liquidity, and charity fees. This centralizes control over the fee structure.
Each type of tax (buy, sell, transfer) can have 0-25% fee.
Code:
function setTaxFeePercent(uint256 taxFeeBps) external onlyOwner { ... }
function setLiquidityFeePercent(uint256 liquidityFeeBps) external onlyOwner { ... }
function setCharityFeePercent(uint256 charityFeeBps) external onlyOwner { ... }

Suggestion:
Ensure that sum of max buy and sell fee is less than 25%
buy + sell fee <= 25%
transfer fee <= 5%

FUNCTIONAL TESTING

Category: Centralization
Subject: Exclusion from fees and rewards
Severity: Informational
Status: not applicable
Overview:
The contract allows the owner to exclude certain addresses from fees and rewards. This centralizes control over the fee and reward distribution.

Code:
function excludeFromReward(address account) public onlyOwner { ... }
function includeInReward(address account) external onlyOwner { ... }
function excludeFromFee(address account) public onlyOwner { ... }

Suggestion:
Consider implementing a decentralized governance mechanism to allow the community to decide on the exclusion or inclusion of addresses in fees and rewards.

FUNCTIONAL TESTING

Category: Centralization
Subject: Swap and liquify settings
Severity: Informational
Status: not applicable
Overview:
The contract allows the owner to set the swap back settings, which affects the swap and liquify process. This centralizes control over the contract's liquidity management.
Setting swap treshold to a large number can increase slippage % on sells
Code:
function setSwapBackSettings(uint256 _amount) external onlyOwner { ... }

Suggestion:
Consider implementing a decentralized governance mechanism to allow the community to decide on the swap back settings and other liquidity management parameters.

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 specializes 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