auditace.tech

AuditAce

FROM INCEPTION TO SUCCESS

Smart Contract Audit

FOR

Milday GPT

DATED : 14 May 23'

AUDIT SUMMARY

Project name – Milady GPT

Date: 14 May, 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	1	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 done on BSC Test network, each test has its transaction has attached to it.

3- Slither : Static Analysis

Testnet Link: all tests were done using this contract, tests are done on BSC Testnet

https://testnet.bscscan.com/token/0xB5366b9945ce2452D8E40d38792cc7b1e307C686

Token Information

Token Name : Milady GPT

Token Symbol: MILADY

Decimals: 9

Token Supply:1,000,000,000,000,000

Token Address:
0xB0583746831508D6c7F436d3f0a4A4C063745b6f

Checksum:
000ec7738d447b0cc7fe7f366fef6f53581689a8

Owner:
0x8Bcce52Ae1eD3fe43Cc0c1C0f881775C95846DdF

TOKEN OVERVIEW

Fees:

Buy Fees: upto 12 %

Sell Fees: upto 12 %

Transfer Fees: 0%

Fees Privilige: owner

Ownership : owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Priviliges: changing swap threshold - changing fees - modifying swap settings

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.

auditace.tech

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

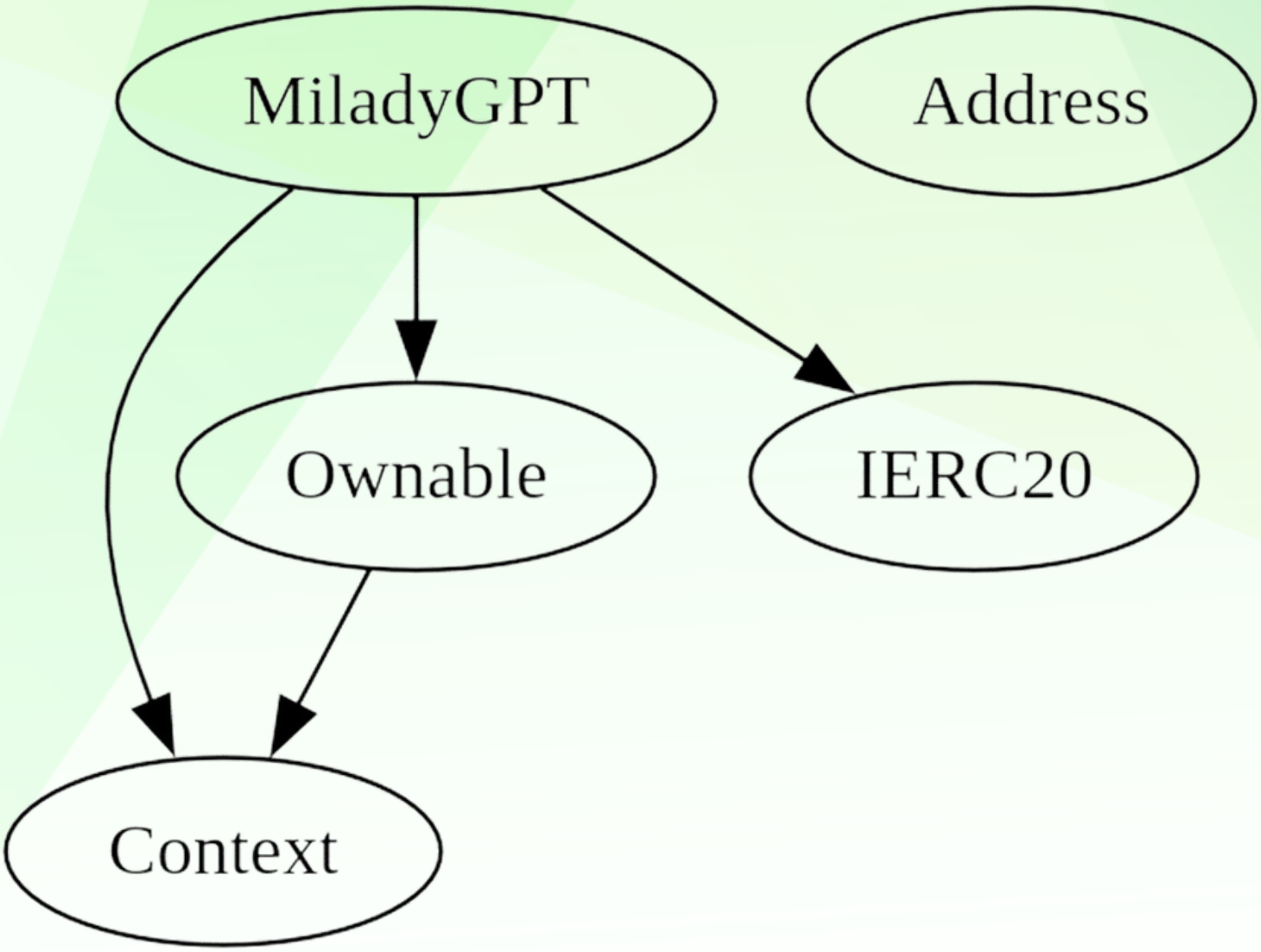
CLASSIFICATION OF RISK

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	0
Low-Risk	1
Gas Optimization / Suggestions	0

INHERITANCE TREE



POINTS TO NOTE

- owner is not able to set buy/sell fees more than 12%
- owner is not able to set transfer fees (0%)
- owner is not able to blacklist an arbitrary wallet
- owner is not able to set limit for buy/sell/transfer/holding amounts
- owner is not able to mint new tokens
- owner is not able to disable trades
- owner can exclude/include an address from fees
- owner can set staking address
- owner can lock/unlock tokens for staking
- owner can update buy/sell fees
- owner can set swap tokens at amount
- owner can enable/disable swap
- owner can claim stuck tokens
- owner can exclude/include an address from rewards
- owner can transfer ownership
- owner can renounce ownership

CONTRACT ASSESMENT

Contract	Type	Bases		
└	**Function Name**	**Visibility**	**Mutability**	**Modifiers**
Context	Implementation			
└	_msgSender	Internal 🔒		
└	_msgData	Internal 🔒		
Ownable	Implementation	Context		
└	<Constructor>	Public ❗	🛑	NO❗
└	owner	Public ❗		NO❗
└	renounceOwnership	Public ❗	🛑	onlyOwner
└	transferOwnership	Public ❗	🛑	onlyOwner
IERC20	Interface			
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	transfer	External ❗	🛑	NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
Address	Library			
└	isContract	Internal 🔒		
└	sendValue	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCall	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	functionCallWithValue	Internal 🔒	🛑	
└	_functionCallWithValue	Private 🔐	🛑	
IUniswapV2Factory	Interface			
└	feeTo	External ❗		NO❗
└	feeToSetter	External ❗		NO❗
└	getPair	External ❗		NO❗
└	allPairs	External ❗		NO❗
└	allPairsLength	External ❗		NO❗
└	createPair	External ❗	🛑	NO❗
└	setFeeTo	External ❗	🛑	NO❗
└	setFeeToSetter	External ❗	🛑	NO❗
IUniswapV2Pair	Interface			

CONTRACT ASSESMENT

└	name	External ❗		NO❗
└	symbol	External ❗		NO❗
└	decimals	External ❗		NO❗
└	totalSupply	External ❗		NO❗
└	balanceOf	External ❗		NO❗
└	allowance	External ❗		NO❗
└	approve	External ❗	🛑	NO❗
└	transfer	External ❗	🛑	NO❗
└	transferFrom	External ❗	🛑	NO❗
└	DOMAIN_SEPARATOR	External ❗		NO❗
└	PERMIT_TYPEHASH	External ❗		NO❗
└	nonces	External ❗		NO❗
└	permit	External ❗	🛑	NO❗
└	MINIMUM_LIQUIDITY	External ❗		NO❗
└	factory	External ❗		NO❗
└	token0	External ❗		NO❗
└	token1	External ❗		NO❗
└	getReserves	External ❗		NO❗
└	price0CumulativeLast	External ❗		NO❗
└	price1CumulativeLast	External ❗		NO❗
└	kLast	External ❗		NO❗
└	burn	External ❗	🛑	NO❗
└	swap	External ❗	🛑	NO❗
└	skim	External ❗	🛑	NO❗
└	sync	External ❗	🛑	NO❗
└	initialize	External ❗	🛑	NO❗
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❗

CONTRACT ASSESMENT

└	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❗
MiladyGPT	Implementation	Context, IERC20, Ownable		
└	<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❗
└	totalReflectionDistributed	Public ❗		NO❗
└	deliver	Public ❗	🛑	NO❗
└	reflectionFromToken	Public ❗		NO❗
└	tokenFromReflection	Public ❗		NO❗
└	excludeFromReward	Public ❗	🛑	onlyOwner
└	includeInReward	External ❗	🛑	onlyOwner
└	<Receive Ether>	External ❗	💸	NO❗
└	claimStuckTokens	External ❗	🛑	NO❗
└	setStakingAddress	External ❗	🛑	onlyOwner
└	lockToken	Public ❗	🛑	NO❗
└	unlockToken	Public ❗	🛑	NO❗
└	updateFeeBuy	Public ❗	🛑	onlyOwner
└	updateFeeSell	Public ❗	🛑	onlyOwner

CONTRACT ASSESMENT

└	_reflectFee	Private 🔐	🛑	
└	_getValues	Private 🔐		
└	_getTValues	Private 🔐		
└	_getRValues	Private 🔐		
└	_getRate	Private 🔐		
└	_getCurrentSupply	Private 🔐		
└	_takeLiquidity	Private 🔐	🛑	
└	_takeMarketing	Private 🔐	🛑	
└	calculateTaxFee	Private 🔐		
└	calculateLiquidityFee	Private 🔐		
└	calculateMarketingFee	Private 🔐		
└	removeAllFee	Private 🔐	🛑	
└	setBuyFee	Private 🔐	🛑	
└	setSellFee	Private 🔐	🛑	
└	isExcludedFromFee	Public ❗		NO❗
└	_approve	Private 🔐	🛑	
└	_transfer	Private 🔐	🛑	
└	swapAndLiquify	Private 🔐	🛑	
└	swapAndSendMarketing	Private 🔐	🛑	
└	setSwapTokensAtAmount	External ❗	🛑	onlyOwner
└	setSwapEnabled	External ❗	🛑	onlyOwner
└	_tokenTransfer	Private 🔐	🛑	
└	_transferStandard	Private 🔐	🛑	
└	_transferToExcluded	Private 🔐	🛑	
└	_transferFromExcluded	Private 🔐	🛑	
└	_transferBothExcluded	Private 🔐	🛑	
└	excludeFromFees	External ❗	🛑	onlyOwner
└	isContract	Internal 🔒		

Legend

Symbol	Meaning
🛑	Function can modify state
💰	Function is payable

STATIC ANALYSIS

```
Variable MiladyGPT._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#940) is too similar to MiladyGPT._getTValues(uint256).tTransferAmount (contracts/
Token.sol#687)
Variable MiladyGPT._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#951) is too similar to MiladyGPT._getValues(uint256).tTransferAmount (contracts/T
oken.sol#678)
Variable MiladyGPT._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#951) is too similar to MiladyGPT._transferFromExcluded(address,address,uint256).t
TransferAmount (contracts/Token.sol#940)
Variable MiladyGPT._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#929) is too similar to MiladyGPT._transferBothExcluded(address,address,uint256).tTr
ansferAmount (contracts/Token.sol#951)
Variable MiladyGPT.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#576) is too similar to MiladyGPT._transferFromExcluded(address,address,uint256).tTransferAmoun
t (contracts/Token.sol#940)
Variable MiladyGPT._transferFromExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#940) is too similar to MiladyGPT._transferToExcluded(address,address,uint256).tTr
ansferAmount (contracts/Token.sol#929)
Variable MiladyGPT._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#929) is too similar to MiladyGPT._getValues(uint256).tTransferAmount (contracts/Tok
en.sol#678)
Variable MiladyGPT._getValues(uint256).rTransferAmount (contracts/Token.sol#679) is too similar to MiladyGPT._transferToExcluded(address,address,uint256).tTransferAmount (contracts/Tok
en.sol#929)
Variable MiladyGPT._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#696) is too similar to MiladyGPT._transferToExcluded(address,address,uint25
6).tTransferAmount (contracts/Token.sol#929)
Variable MiladyGPT._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#929) is too similar to MiladyGPT._transferStandard(address,address,uint256).tTransf
erAmount (contracts/Token.sol#919)
Variable MiladyGPT.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#576) is too similar to MiladyGPT._transferBothExcluded(address,address,uint256).tTransferAmoun
t (contracts/Token.sol#951)
Variable MiladyGPT._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#919) is too similar to MiladyGPT._transferToExcluded(address,address,uint256).tTransf
erAmount (contracts/Token.sol#929)
Variable MiladyGPT._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#929) is too similar to MiladyGPT._transferToExcluded(address,address,uint256).tTran
sferAmount (contracts/Token.sol#929)
Variable MiladyGPT._getRValues(uint256,uint256,uint256,uint256,uint256).rTransferAmount (contracts/Token.sol#696) is too similar to MiladyGPT._transferFromExcluded(address,address,uint
256).tTransferAmount (contracts/Token.sol#940)
Variable MiladyGPT._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#919) is too similar to MiladyGPT._transferFromExcluded(address,address,uint256).tTran
sferAmount (contracts/Token.sol#940)
Variable MiladyGPT._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#929) is too similar to MiladyGPT._getTValues(uint256).tTransferAmount (contracts/To
ken.sol#687)
Variable MiladyGPT._transferBothExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#951) is too similar to MiladyGPT._transferToExcluded(address,address,uint256).tTr
ansferAmount (contracts/Token.sol#929)
Variable MiladyGPT._transferToExcluded(address,address,uint256).rTransferAmount (contracts/Token.sol#929) is too similar to MiladyGPT._transferFromExcluded(address,address,uint256).tTr
ansferAmount (contracts/Token.sol#940)
Variable MiladyGPT._transferStandard(address,address,uint256).rTransferAmount (contracts/Token.sol#919) is too similar to MiladyGPT._transferBothExcluded(address,address,uint256).tTran
sferAmount (contracts/Token.sol#951)
Variable MiladyGPT.reflectionFromToken(uint256,bool).rTransferAmount (contracts/Token.sol#576) is too similar to MiladyGPT._transferToExcluded(address,address,uint256).tTransferAmount
(contracts/Token.sol#929)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

MiladyGPT.DEAD (contracts/Token.sol#412) should be constant
MiladyGPT._decimals (contracts/Token.sol#386) should be constant
MiladyGPT._name (contracts/Token.sol#384) should be constant
MiladyGPT._symbol (contracts/Token.sol#385) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

MiladyGPT.DEV (contracts/Token.sol#413) should be immutable
MiladyGPT._tTotal (contracts/Token.sol#389) should be immutable
MiladyGPT.mk (contracts/Token.sol#409) should be immutable
MiladyGPT.mkTwo (contracts/Token.sol#410) should be immutable
MiladyGPT.totalBuyFees (contracts/Token.sol#406) should be immutable
MiladyGPT.totalSellFees (contracts/Token.sol#407) should be immutable
MiladyGPT.uniswapV2Pair (contracts/Token.sol#416) should be immutable
MiladyGPT.uniswapV2Router (contracts/Token.sol#415) 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

All the functionalities have been tested, no issues were found

1- Adding liquidity (passed):
https://testnet.bscscan.com/tx/0xd3330960eac0542a7a14eb28f90fcb036d39b9365ab01adc5beb82beb5446fa4

2- Buying when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0xe5c4116886b50b6c5a5db207548099fe1d30acbcfd781a891f7d1ca0831d9c0a

3- Selling when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x0519eaed77a74559d0477f22266c71cfdf87d6b2f9f7026c965b3e17a4252136

4- Transferring when excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x92aa62d2f21bf35c099e387d4405b9959c7ee1c64bfb190a73cf61f8b5de3bdc

5- Buying when not excluded (0-12% tax) (passed):
https://testnet.bscscan.com/tx/0xbdac56f2bc19c0a837a445ac7ca8c693423f53c48982bc89581a4f83249b3244

6- Selling when not excluded (0- 12% tax) (passed):
https://testnet.bscscan.com/tx/0x62740621c344507d0444fff63ea101e68b91c7660431029754ae369cb96c23aa

FUNCTIONAL TESTING

7- Transferring when not excluded (0% tax) (passed):
https://testnet.bscscan.com/tx/0x0e69d959be5356353c8f318b03be3c1b40eeaead8a71ec764840b3781bdc74f0

8- Auto liquidity, Marketing fee (passed):
https://testnet.bscscan.com/tx/0xd1b2bf21dd2af8d751b7aeb54119d304ddde02b86fd053c98174dcc4640de8b3

MANUAL TESTING

Logical – Large swap threshold can increase slippage

Severity: Low

function: setSwapTokensAtAmount

Status: Not Resolved

Overview:

Owner is able to set swapTokensAtAmount to a large number, this can increase slippage up to 49% if contract balance is more than swapTokensAtAmount

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner() {
    require(newAmount > totalSupply() / 1e5, "SwapTokensAtAmount must be greater than
        0.001% of total supply");
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(newAmount);
}
```

Suggestion

To mitigate this Logical issue, make sure that swapTokensAtAmount is always less than 1% of supply

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner() {
    require(newAmount > totalSupply() / 1e5, "SwapTokensAtAmount must be greater than
0.001% of total supply");
    require(newAmount < totalSupply() / 1e2, "SwapTokensAtAmount must be greater than
0.001% of total supply");
    swapTokensAtAmount = newAmount;
    emit SwapTokensAtAmountUpdated(newAmount);
}
```

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