



Smart Contract Audit

FOR
Milady
DATED : 12 May 23'



AUDIT SUMMARY

Project name - MILADY

Date: 12 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	1	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 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/address/0xfb955D15DAdF2e8C31978fA5aC32D6fa2396B323#code>



Token Information

Token Name : Milady

Token Symbol: MILADY

Decimals: 18

Token Supply: 1,000,000,000

Token Address:

0x26Ca3F140c9cC2a44fafA8Ab669034b3393d6e29

Checksum:

0e68c31fba552480c7a519b9e8d042d67be0ae22

Owner:

0xbcF948759BbEe8Bed05ee47EBd28cF762a77a5E5

Deployer:

0xbcF948759BbEe8Bed05ee47EBd28cF762a77a5E5



TOKEN OVERVIEW

Fees:

Buy Fees: 0%

Sell Fees: 0%

Transfer Fees: 0%

Fees Privilege: None

Ownership: Owned

Minting: No mint function

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: enabling trades



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 Limit and 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



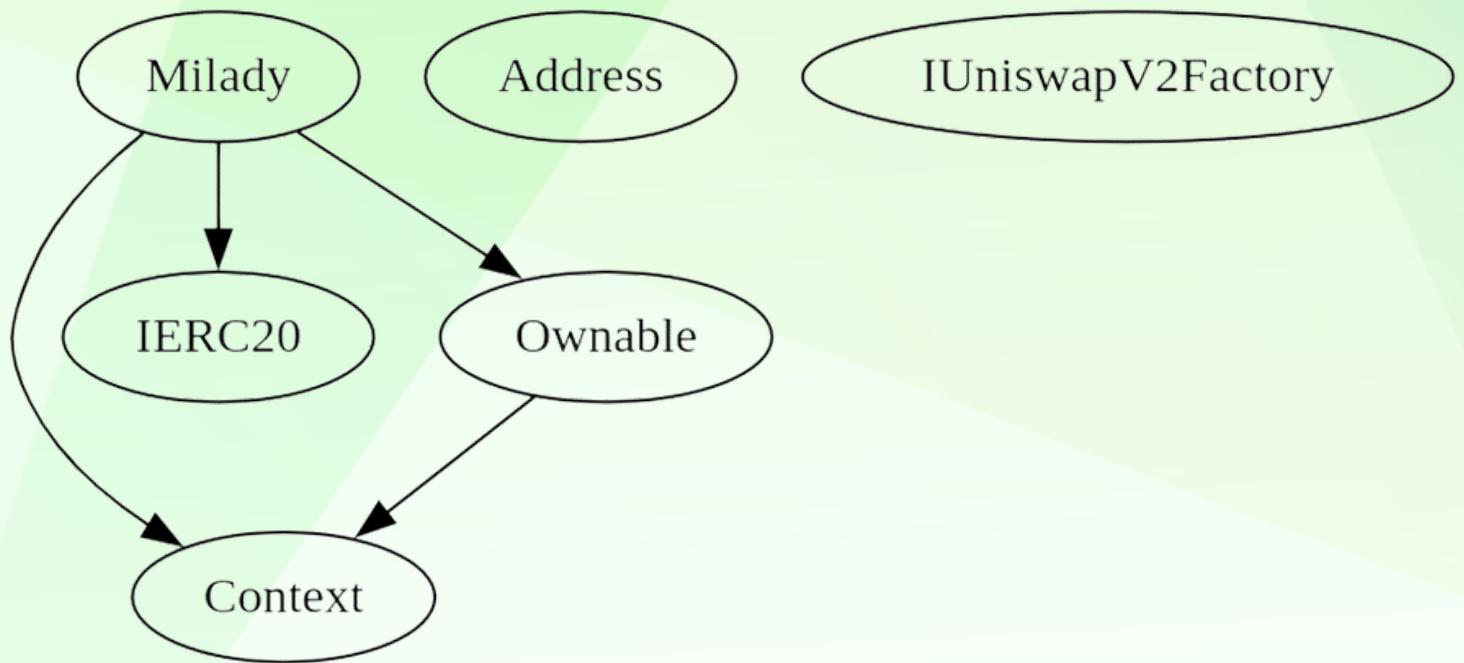
CLASSIFICATION OF RISK

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	1
◆ Low-Risk	1
◆ Gas Optimization / Suggestions	0

INHERITANCE TREE





POINTS TO NOTE

- Owner is not able to set fees (0%)
- Owner is not able to set max buy/sell/transfer/hold amount
- Owner is not able to blacklist an arbitrary wallet
- Owner is not able to disable trades
- Owner is not able to mint new tokens
- **Owner must enable trading for investors**



CONTRACT ASSESSMENT

Contract	Type	Bases			
L	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	
Context	Implementation				
L	_msgSender	Internal	🔓		
L	_msgData	Internal	🔓		
IERC20	Interface				
L	totalSupply	External	!	NO !	
L	balanceOf	External	!	NO !	
L	transfer	External	!	🔒	NO !
L	allowance	External	!	NO !	
L	approve	External	!	🔒	NO !
L	transferFrom	External	!	🔒	NO !
Address	Library				
L	isContract	Internal	🔓		
L	sendValue	Internal	🔓	🔒	
L	functionCall	Internal	🔓	🔒	
L	functionCall	Internal	🔓	🔒	
L	functionCallWithValue	Internal	🔓	🔒	
L	functionCallWithValue	Internal	🔓	🔒	
L	functionStaticCall	Internal	🔓		
L	functionStaticCall	Internal	🔓		
L	functionDelegateCall	Internal	🔓	🔒	
L	functionDelegateCall	Internal	🔓	🔒	
L	verifyCallResultFromTarget	Internal	🔓		
L	verifyCallResult	Internal	🔓		

CONTRACT ASSESSMENT

| L | _revert | Private  | ||

|||||

| **Ownable** | Implementation | Context |||

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

| L | owner | Public ! | | NO ! |

| L | _checkOwner | Internal  | ||

| L | renounceOwnership | Public ! |  | onlyOwner |

| L | transferOwnership | Public ! |  | onlyOwner |

| L | _transferOwnership | Internal  |  | |

|||||

| **IUniswapV2Factory** | Interface | |||

| L | feeTo | External ! | | NO ! |

| L | feeToSetter | External ! | | NO ! |

| L | getPair | External ! | | NO ! |

| L | allPairs | External ! | | NO ! |

| L | allPairsLength | External ! | | NO ! |

| L | createPair | External ! |  | NO ! |

| L | setFeeTo | External ! |  | NO ! |

| L | setFeeToSetter | External ! |  | NO ! |

|||||

| **IUniswapV2Pair** | Interface | |||

| L | name | External ! | | NO ! |

| L | symbol | External ! | | NO ! |

| L | decimals | External ! | | NO ! |

| L | totalSupply | External ! | | NO ! |

| L | balanceOf | External ! | | NO ! |

| L | allowance | External ! | | NO ! |



CONTRACT ASSESSMENT

```
| L | approve | External ! |  | NO ! | |
| L | transfer | External ! |  | NO ! |  
| L | transferFrom | External ! |  | NO ! |  
| L | DOMAIN_SEPARATOR | External ! | | NO ! |  
| L | PERMIT_TYPEHASH | External ! | | NO ! |  
| L | nonces | External ! | | NO ! |  
| L | permit | External ! |  | NO ! |  
| L | MINIMUM_LIQUIDITY | External ! | | NO ! |  
| L | factory | External ! | | NO ! |  
| L | token0 | External ! | | NO ! |  
| L | token1 | External ! | | NO ! |  
| L | getReserves | External ! | | NO ! |  
| L | price0CumulativeLast | External ! | | NO ! |  
| L | price1CumulativeLast | External ! | | NO ! |  
| L | kLast | External ! | | NO ! |  
| L | burn | External ! |  | NO ! |  
| L | swap | External ! |  | NO ! |  
| L | skim | External ! |  | NO ! |  
| L | sync | External ! |  | NO ! |  
| L | initialize | External ! |  | NO ! |  
|||||  
| **IUniswapV2Router01** | Interface | |||  
| L | factory | External ! | | NO ! |  
| L | WETH | External ! | | NO ! |  
| L | addLiquidity | External ! |  | NO ! |  
| L | addLiquidityETH | External ! | |  | NO ! |  
| L | removeLiquidity | External ! |  | NO ! |
```



CONTRACT ASSESSMENT

| L | removeLiquidityETH | External ! | | NO ! |

| L | removeLiquidityWithPermit | External ! | | NO ! |

| L | removeLiquidityETHWithPermit | External ! | | NO ! |

| L | swapExactTokensForTokens | External ! | | NO ! |

| L | swapTokensForExactTokens | External ! | | NO ! |

| L | swapExactETHForTokens | External ! | | NO ! |

| L | swapTokensForExactETH | External ! | | NO ! |

| L | swapExactTokensForETH | External ! | | NO ! |

| L | swapETHForExactTokens | External ! | | NO ! |

| L | quote | External ! | | NO ! |

| L | getAmountOut | External ! | | NO ! |

| L | getAmountIn | External ! | | NO ! |

| L | getAmountsOut | External ! | | NO ! |

| L | getAmountsIn | External ! | | NO ! |

|||||

| **IUniswapV2Router02** | Interface | IUniswapV2Router01 |||

| L | removeLiquidityETHSupportingFeeOnTransferTokens | External ! | | NO ! |

| L | removeLiquidityETHWithPermitSupportingFeeOnTransferTokens | External ! | | NO ! |

| L | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! | | NO ! |

| L | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! | | NO ! |

| L | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! | | NO ! |

|||||

| **Milady** | Implementation | Context, IERC20, Ownable |||

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

| L | totalSupply | Public ! | | NO ! |

| L | balanceOf | Public ! | | NO ! |

| L | transfer | Public ! | | NO ! |

CONTRACT ASSESSMENT

```

| L | allowance | Public ! | NO ! | |
| L | approve | Public ! |  NO ! |
| L | transferFrom | Public ! |  NO ! |
| L | increaseAllowance | Public ! |  NO ! |
| L | decreaseAllowance | Public ! |  NO ! |
| L | _approve | Private  |  ||
| L | _transfer | Private  |  ||
| L | swapAndLiquify | Public ! |  | lockTheSwap |
| L | swapTokensForEth | Private  |  ||
| L | _tokenTransfer | Private  |  ||
| L | isExcludedFromFee | External ! | NO ! |
| L | excludeFromFee | External ! |  | onlyOwner |
| L | includeInFee | External ! |  | onlyOwner |
| L | setTokensToSwap | External ! |  | onlyOwner |
| L | setSwapAndLiquifyEnabled | External ! |  | onlyOwner |
| L | setMarketingWallet | External ! |  | onlyOwner |
| L | transferToAddressETH | Private  |  ||
| L | <Receive Ether> | External ! |  NO ! |
| L | swapETHForTokens | Private  |  ||
| L | recoverETHfromContract | External ! |  | onlyOwner |
| L | recoverTokensFromContract | External ! |  | onlyOwner |
| L | enableTrading | External ! |  | onlyOwner |
|||||
| **DividendDistributor** | Implementation | IDividendDistributor ||
| L | <Constructor> | Public ! |  NO ! |
| L | setDistributionCriteria | External ! |  | onlyToken |
| L | setShare | External ! |  | onlyToken |

```



CONTRACT ASSESSMENT

```
| L | deposit | External ! | ☰ | onlyToken | |
| L | process | External ! | ☰ | onlyToken |
| L | shouldDistribute | Internal ☰ | ||
| L | distributeDividend | Internal ☰ | ☰ |||
| L | claimDividend | External ! | ☰ | onlyToken |
| L | getUnpaidEarnings | Public ! | |NO ! |
| L | getCumulativeDividends | Internal ☰ | ||
| L | addShareholder | Internal ☰ | ☰ |||
| L | removeShareholder | Internal ☰ | ☰ |||
| L | setDividendTokenAddress | External ! | ☰ | onlyToken |
|||||||
| **FARMPEPE** | Implementation | IBEP20, Auth |||
| L | <Constructor> | Public ! | ☰ | Auth |
| L | <Receive Ether> | External ! | ☰ | NO ! |
| L | totalSupply | External ! | |NO ! |
| L | decimals | External ! | |NO ! |
| L | symbol | External ! | |NO ! |
| L | name | External ! | |NO ! |
| L | getOwner | External ! | |NO ! |
| L | balanceOf | Public ! | |NO ! |
| L | allowance | External ! | |NO ! |
| L | approve | Public ! | ☰ | NO ! |
| L | approveMax | External ! | ☰ | NO ! |
| L | savetokens | External ! | ☰ | devwall |
| L | burning | External ! | ☰ | NO ! |
| L | transfer | External ! | ☰ | NO ! |
| L | transferFrom | External ! | ☰ | NO ! |
```



CONTRACT ASSESSMENT

```

| L | _transferFrom | Internal 🔒 | 🔒 ||

| L | _basicTransfer | Internal 🔒 | 🔒 ||

| L | shouldTakeFee | Internal 🔒 | ||

| L | shouldTakeFee | Internal 🔒 | ||

| L | getTotalFee | Public ! | |NO ! |

| L | getMultipliedFee | Public ! | |NO ! |

| L | takeFee | Internal 🔒 | 🔒 ||

| L | shouldSwapBack | Internal 🔒 | ||

| L | swapBack | Internal 🔓 | 🔒 | swapping |

| L | buyTokens | Internal 🔒 | 🔒 | swapping |

| L | launched | Internal 🔒 | ||

| L | setFeeReceivers | External ! | 🔒 | devwall |

| L | setSwapBackSettings | External ! | 🔒 | devwall |

| L | setTargetLiquidity | External ! | 🔒 | onlyOwner |

| L | manualSend | External ! | 🔒 | NO ! |

| L | setDistributionCriteria | External ! | 🔒 | onlyOwner |

| L | claimDividend | External ! | 🔒 | NO ! |

| L | getUnpaidEarnings | Public ! | |NO ! |

| L | setDistributorSettings | External ! | 🔒 | onlyOwner |

| L | getCirculatingSupply | Public ! | |NO ! |

| L | getLiquidityBacking | Public ! | |NO ! |

| L | isOverLiquified | Public ! | |NO ! |

||||||

| **Auth** | Implementation | ||

| L | <Constructor> | Public ! | 🔒 | NO ! |

| L | isOwner | Public ! | |NO ! |

| L | isdevwallet | Public ! | |NO ! |

```



CONTRACT ASSESSMENT

| L | transferOwnership | Public ! | | onlyOwner |

| L | renounceOwnership | Public ! | | onlyOwner |

|||||

| **IERC165** | Interface | |||

| L | supportsInterface | External ! | |NO ! |

|||||

| **IBEP20** | Interface | |||

| L | totalSupply | External ! | |NO ! |

| L | decimals | External ! | |NO ! |

| L | symbol | External ! | |NO ! |

| L | name | External ! | |NO ! |

| L | getOwner | External ! | |NO ! |

| L | balanceOf | External ! | |NO ! |

| L | transfer | External ! | | |NO ! |

| L | burning | External ! | | |NO ! |

| L | allowance | External ! | |NO ! |

| L | approve | External ! | | |NO ! |

| L | transferFrom | External ! | | |NO ! |

|||||

| **IDEXFactory** | Interface | |||

| L | createPair | External ! | | |NO ! |

|||||

| **IDEXRouter** | Interface | |||

| L | factory | External ! | |NO ! |

| L | WETH | External ! | |NO ! |

| L | addLiquidity | External ! | | |NO ! |

| L | addLiquidityETH | External ! | | |NO ! |



CONTRACT ASSESSMENT

L swapExactTokensForTokensSupportingFeeOnTransferTokens External !	NO !	○
L swapExactETHForTokensSupportingFeeOnTransferTokens External !	NO !	⚡
L swapExactTokensForETHSupportingFeeOnTransferTokens External !	NO !	○
SafeMath Library		
L tryAdd Internal 🔒		
L trySub Internal 🔒		
L tryMul Internal 🔒		
L tryDiv Internal 🔒		
L tryMod Internal 🔒		
L add Internal 🔒		
L sub Internal 🔒		
L mul Internal 🔒		
L div Internal 🔒		
L mod Internal 🔒		
L sub Internal 🔒		
L div Internal 🔒		
L mod Internal 🔒		
IDividendDistributor Interface		
L setDistributionCriteria External !	NO !	○
L setShare External !	NO !	○
L deposit External !	NO !	⚡
L process External !	NO !	○
Symbol Meaning		
:-----: -----		
○ Function can modify state		
⚡ Function is payable		

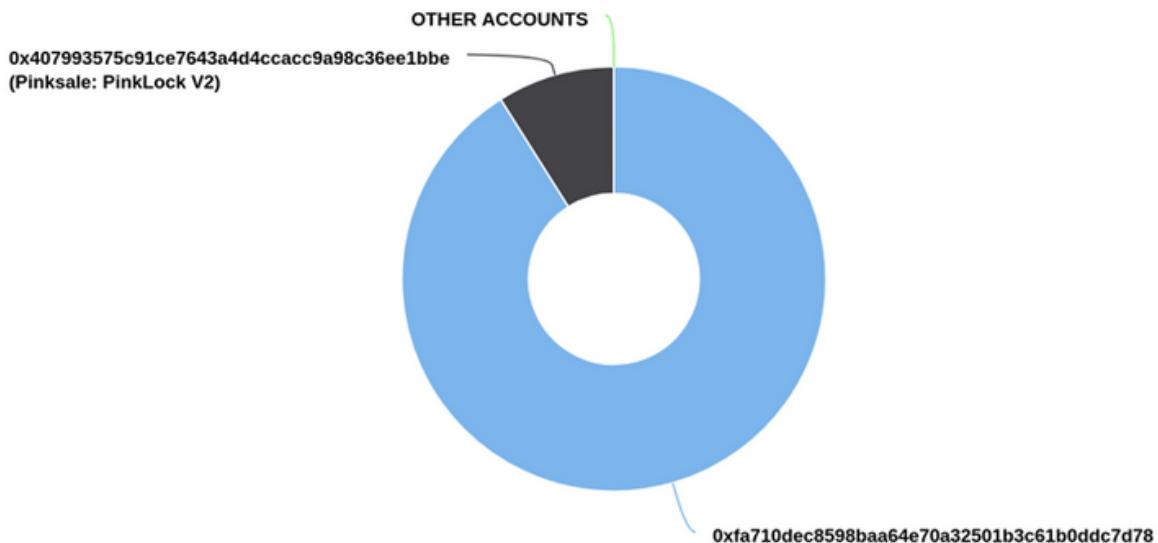
TOKEN DISTRIBUTION AT TIME OF AUDIT

💡 The top 100 holders collectively own 100.00%
(420,690,000,000,000.00 Tokens) of Milady

💡 Token Total Supply: 420,690,000,000,000.00 Token | Total
Token Holders: 2

Milady Top 100 Token Holders

Source: BscScan.com





STATIC ANALYSIS

```
Low level call in Address.sendValue(address,uint256) (contracts/Token.sol#102-113):
- (success) = recipient.call{value: amount}() (contracts/Token.sol#108)
Low level call in Address.functionCallWithValue(address,bytes,uint256,string) (contracts/Token.sol#191-211):
- (success,returndata) = target.call{value: value}(data) (contracts/Token.sol#201-203)
Low level call in Address.functionStaticcall(address,bytes,string) (contracts/Token.sol#237-250):
- (success,returndata) = target.staticcall(data) (contracts/Token.sol#242)
Low level call in Address.functionDelegatecall(address,bytes,string) (contracts/Token.sol#276-289):
- (success,returndata) = target.delegatecall(data) (contracts/Token.sol#281)
Low level call in Milady.transferToAddressETH(address,uint256) (contracts/Token.sol#1108-1114):
- (succ) = recipient.call{value: amount}() (contracts/Token.sol#1112)
Low level call in Milady.recoverETHfromContract() (contracts/Token.sol#1141-1149):
- (succ) = address(marketingWallet).call{value: ethBalance}() (contracts/Token.sol#1143)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#low-level-calls

Function IUniswapV2Pair.DOMAIN_SEPARATOR() (contracts/Token.sol#498) is not in mixedCase
Function IUniswapV2Pair.PERMIT_TYPEHASH() (contracts/Token.sol#500) is not in mixedCase
Function IUniswapV2Pair.MINIMUM_LIQUIDITY() (contracts/Token.sol#530) is not in mixedCase
Function IUniswapV2Router01.WETH() (contracts/Token.sol#570) is not in mixedCase
Parameter Milady.allowance(address,address) . owner (contracts/Token.sol#899) is not in mixedCase
Parameter Milady.setTokensToSwap(uint256) . minimumTokensBeforeSwap (contracts/Token.sol#1082) is not in mixedCase
Parameter Milady.setSwapAndLiquifyEnabled(bool) . _enabled (contracts/Token.sol#1095) is not in mixedCase
Parameter Milady.setMarketingWallet(address) . _marketingWallet (contracts/Token.sol#1102) is not in mixedCase
Parameter Milady.recoverTokensFromContract(address,uint256) . _tokenAddress (contracts/Token.sol#1153) is not in mixedCase
Parameter Milady.recoverTokensFromContract(address,uint256) . amount (contracts/Token.sol#1154) is not in mixedCase
Constant Milady.contractVersion (contracts/Token.sol#766) is not in UPPER_CASE_WITH_UNDERSCORES
Constant Milady.contractDev (contracts/Token.sol#767) is not in UPPER_CASE_WITH_UNDERSCORES
Constant Milady.contractEdition (contracts/Token.sol#768) is not in UPPER_CASE_WITH_UNDERSCORES
Constant Milady.deadWallet (contracts/Token.sol#777-778) is not in UPPER_CASE_WITH_UNDERSCORES
Variable Milady.WETH (contracts/Token.sol#818) is not in mixedCase
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#conformance-to-solidity-naming-conventions

Variable IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountDesired (contracts/Token.sol#575) is too similar to IUniswapV2Router01.addLiquidity(address,address,uint256,uint256,uint256,uint256,address,uint256).amountBDesired (contracts/Token.sol#576)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#variable-names-too-similar

Milady._tFeeTotal (contracts/Token.sol#796) is never used in Milady (contracts/Token.sol#763-1176)
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#unused-state-variable

Milady._tFeeTotal (contracts/Token.sol#796) should be constant
Milady._tTotal (contracts/Token.sol#795) should be constant
Milady.buyFee (contracts/Token.sol#802) should be constant
Milady.sellFee (contracts/Token.sol#804) should be constant
Milady.swapOutput (contracts/Token.sol#812) should be constant
Reference: https://github.com/crytic/slither/wiki/Detector-Documentation#state-variables-that-could-be-declared-constant

Milady.safuWallet (contracts/Token.sol#773-774) 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/0x151f241df274e95d71e70c51fe5c57c044d7a566f3f9c6e86f885d269aa44cff>

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

<https://testnet.bscscan.com/tx/0xe7edb45c367aaa02ad49a8d5e2dcc08e418d171cb1b47ac5957deb43e43c4014>

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

<https://testnet.bscscan.com/tx/0x41a334aa0d06e39cb934d13f44068f7187efece1af704f16345a5d40abf886fc>

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

<https://testnet.bscscan.com/tx/0x67692a3189839738378e9c87c16e83504b5d4ddc83a59b1de292dc4921c849d1>

5- Buying when not excluded from fees (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x0a0e2a4fd95599ab022830932dc14a49f33397df971022dd390f42b988731e40>

6- Selling when not excluded from fees (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0x84d36c9f2e40d77cc5927e9872905424619e7bb>
[a69de3a1a9c6630fb9e6d7c1d](https://testnet.bscscan.com/tx/0xa69de3a1a9c6630fb9e6d7c1d)

7- Transferring when not excluded from fees (0% tax) (**passed**):

<https://testnet.bscscan.com/tx/0xf2ec21a0a525d436c3b5aa4d93e18d7524c4aeef>
[9aa8c5be833181773f9af2f6](https://testnet.bscscan.com/tx/0x9aa8c5be833181773f9af2f6)



MANUAL TESTING

Logical – Trades must be enabled before adding liquidity and claiming tokens

Severity: Medium

function: enableTrading

Status: Not Resolved

Overview:

if trades are not enabled, no one would be able to transfer/buy/sell tokens (even owner of the contract)

this means owner has to enable trading before adding liquidity or finalizing presale which removes the whole point of having an enable trading function.

```
function _transfer(address from, address to, uint256 amount) private {
    require(from != address(0), "ERC20: transfer from the zero address");
    require(to != address(0), "ERC20: transfer to the zero address");
    require(amount > 0, "Transfer amount must be greater than zero");
    require(
        _tOwned[from] >= amount,
        "ERC20: transfer amount exceeds balance"
    );
    //Enable Trade after sale is finalized
    require(tradingEnabled, "Trading not yet enabled!");
```

Suggestion

To mitigate this issue, declare a mapping to keep track of whitelisted wallets, this whitelisted wallets would be able to transfer tokens even if trades are disabled

```
function _transfer(address from, address to, uint256 amount) private {
    require(from != address(0), "ERC20: transfer from the zero address");
    require(to != address(0), "ERC20: transfer to the zero address");
    require(amount > 0, "Transfer amount must be greater than zero");
    require(
        _tOwned[from] >= amount,
        "ERC20: transfer amount exceeds balance"
    );
    if(!whitelisted[from] && !whitelisted[to] {
        require(tradingEnabled, "Trading not yet enabled!");
    }
```



MANUAL TESTING

Logical – Tax is always 0

Severity: Low

Status: Not Resolved

Overview:

Although token has the implementation for charging fees and perform an internal sap, Current fees are set to 0 and owner is not able to change it later.

Suggestion

It's suggested to either remove the sections of the code that are related to fees, or implement a function to be able to change fees within a safe range (0-25% total buy + sell)



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>
