



Smart Contract Audit

FOR

Milady GPT

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	2	0	2	0	1
Acknowledged	0	0	0	0	0
Resolved	0	0	0	0	0

USED TOOLS

Tools:

1.Manual Review: The code has undergone a line-by-line review by the **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:

Contract has been tested on binance smart chain testnet which can be found in below link:

<https://testnet.bscscan.com/address/0x2eD03d4609F9FE2Dfb3a7BE25615E21D80bEDF97>



Token Information

Name : MiladyGPT

Symbol : MILADY

Decimals: 18

Network: Binance smart chain

Token Type: BEP20

Token Address :

0xEB90025B6f7f9260a0b32dcB9eD1A63C70eB3Fc7

Owner:

0x23E95c782F2a6a63A89ff9Cb15f6e3a258Ef55f2

(at time of writing the audit)

Deployer:0x23E95c782F2a6a63A89ff9Cb15f6e3a258Ef55f2



Token Information

Fees:

Buy Fees: up to 25%

Sell Fees: up to 25%

Transfer Fees: up to 25%

Fees Privilege: Owner

Ownership : Owned

Minting: None

Max Tx Amount/ Max Wallet Amount: No

Blacklist: No

Other Privileges: Changing swap threshold - changing fees - modifying buyback 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 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

2

◆ High-Risk

0

◆ Medium-Risk

2

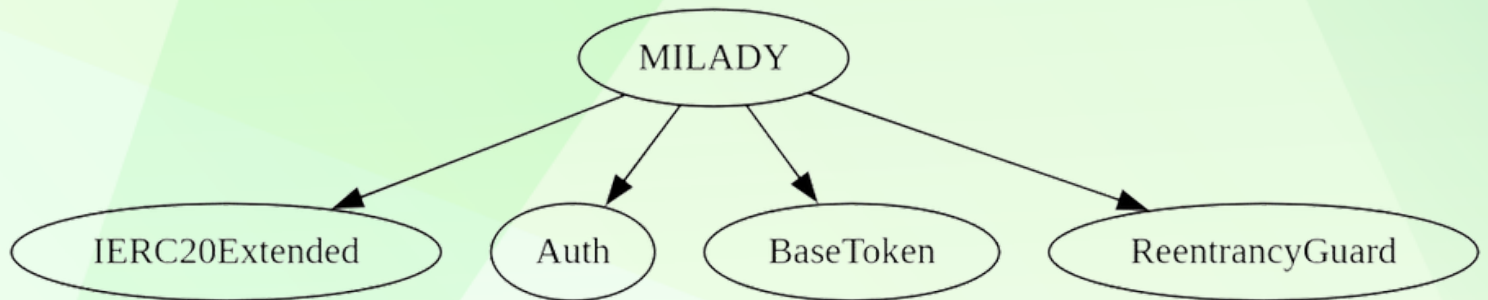
◆ Low-Risk

0

◆ Gas Optimization / Suggestions

1

INHERITANCE TREE





POINTS TO NOTE

- Owner is able to set each type of tax to 25% max
 - 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
-



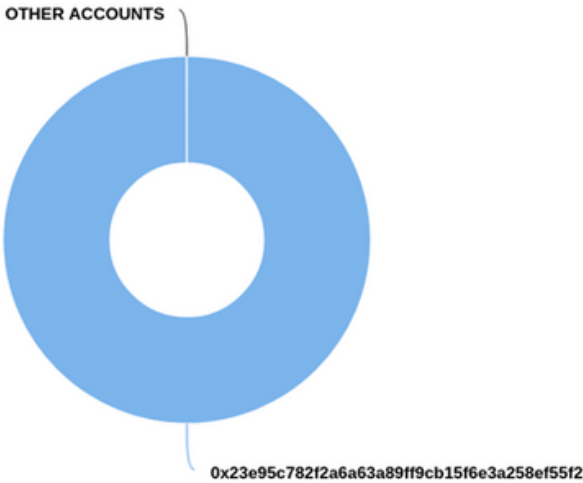
TOKENOMICS AT TIME OF AUDIT

💡 The top 100 holders collectively own 100.00%
(1,000,000,000,000,000.00 Tokens) of MiladyGPT

💡 Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 1

MiladyGPT Top 100 Token Holders

Source: BscScan.com





CONTRACT ASSESMENT

Contract	Type	Bases			
↳	**Function Name**	**Visibility**	**Mutability**	**Modifiers**	
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 🔒			
Clones	Library				
↳	clone	Internal 🔒		🚫	
↳	cloneDeterministic	Internal 🔒		🚫	
↳	predictDeterministicAddress	Internal 🔒			
↳	predictDeterministicAddress	Internal 🔒			
IUniswapV2Factory	Interface				
↳	feeTo	External !		NO !	
↳	feeToSetter	External !		NO !	
↳	getPair	External !		NO !	

CONTRACT ASSESMENT

```
|  | allPairs | External  | | | NO  | |
|  | allPairsLength | External  | | | NO  | |
|  | createPair | External  | |  | NO  | |
|  | setFeeTo | External  | |  | NO  | |
|  | setFeeToSetter | 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  | |
|  | swapETHForExactTokens | External  | |  | NO  | |
|  | quote | External  | | | NO  | |
|  | getAmountOut | External  | | | NO  | |
|  | getAmountIn | External  | | | NO  | |
|  | getAmountsOut | External  | | | NO  | |
```



CONTRACT ASSESMENT

| ^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 ! |

|||||

| **IERC20Extended** | Interface | |||

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

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

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

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

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

| ^L | transfer | External ! | ● | NO ! |

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

| ^L | approve | External ! | ● | NO ! |

| ^L | transferFrom | External ! | ● | NO ! |

|||||

| **Auth** | Implementation | |||

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

| ^L | authorize | Public ! | ● | onlyOwner |

| ^L | unauthorize | Public ! | ● | onlyOwner |

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

| ^L | isAuthorized | Public ! | | NO ! |

CONTRACT ASSESMENT

| ^L | transferOwnership | Public ! | ● | onlyOwner |

|||||

| **IDividendDistributor** | Interface | |||

| ^L | setDistributionCriteria | External ! | ● | NO ! |

| ^L | setShare | External ! | ● | NO ! |

| ^L | deposit | External ! | ⚠ | NO ! |

| ^L | process | External ! | ● | NO ! |

|||||

| **DividendDistributor** | Implementation | IDividendDistributor |||

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

| ^L | setDistributionCriteria | External ! | ● | onlyToken |

| ^L | setShare | External ! | ● | onlyToken |

| ^L | deposit | External ! | ⚠ | onlyToken |

| ^L | process | External ! | ● | onlyToken |

| ^L | shouldDistribute | Internal 🔒 | | |

| ^L | distributeDividend | Internal 🔒 | ● | |

| ^L | claimDividend | External ! | ● | NO ! |

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

| ^L | getCumulativeDividends | Internal 🔒 | | |

| ^L | addShareholder | Internal 🔒 | ● | |

| ^L | removeShareholder | Internal 🔒 | ● | |

|||||

| **BaseToken** | Implementation | |||

|||||

| **ReentrancyGuard** | Implementation | |||

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



CONTRACT ASSESMENT

|||||

| **MILADY** | Implementation | IERC20Extended, Auth, BaseToken, ReentrancyGuard |||

| ^L | <Constructor> | Public ! | ● | Auth |

| ^L | _initializeFees | Internal 🔒 | ● | |

| ^L | _initializeLiquidityBuyBack | Internal 🔒 | ● | |

| ^L | <Receive Ether> | External ! | 📄 | NO ! |

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

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

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

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

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

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

| ^L | approve | Public ! | ● | NO ! |

| ^L | approveMax | External ! | ● | NO ! |

| ^L | transfer | External ! | ● | NO ! |

| ^L | transferFrom | External ! | ● | NO ! |

| ^L | _transferFrom | Internal 🔒 | ● | |

| ^L | _basicTransfer | Internal 🔒 | ● | |

| ^L | shouldTakeFee | Internal 🔒 | | |

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

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

| ^L | takeFee | Internal 🔒 | ● | |

| ^L | shouldSwapBack | Internal 🔒 | | |

| ^L | swapBack | Internal 🔒 | ● | swapping |

| ^L | shouldAutoBuyback | Internal 🔒 | | |



CONTRACT ASSESMENT

| ^L | triggerZeusBuyback | External ! | ● | authorized |

| ^L | claimStuckTokens | External ! | ● | nonReentrant |

| ^L | clearBuybackMultiplier | External ! | ● | authorized |

| ^L | triggerAutoBuyback | Internal 🔒 | ● | |

| ^L | buyTokens | Internal 🔒 | ● | swapping |

| ^L | setAutoBuybackSettings | External ! | ● | authorized |

| ^L | setBuybackMultiplierSettings | External ! | ● | authorized |

| ^L | setIsDividendExempt | External ! | ● | authorized |

| ^L | setIsFeeExempt | External ! | ● | authorized |

| ^L | setBuyBacker | External ! | ● | authorized |

| ^L | setFees | Public ! | ● | authorized |

| ^L | _setFees | Internal 🔒 | ● | |

| ^L | setFeeReceivers | External ! | ● | authorized |

| ^L | setSwapBackSettings | External ! | ● | authorized |

| ^L | setTargetLiquidity | External ! | ● | authorized |

| ^L | setDistributionCriteria | External ! | ● | authorized |

| ^L | setDistributorSettings | External ! | ● | authorized |

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

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

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

|||||


| ****DividendDistributor**** | Implementation | IDividendDistributor |||


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


| ^L | setDistributionCriteria | External ! | ● | onlyToken |



| ^L | setShare | External ! | ● | onlyToken |


CONTRACT ASSESMENT

| ^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 ! |



CONTRACT ASSESMENT

| ^L | transfer | External ! | ● | NO ! |

| ^L | transferFrom | External ! | ● | NO ! |

| ^L | _transferFrom | Internal 🔒 | ● | |

| ^L | _basicTransfer | Internal 🔒 | ● | |

| ^L | shouldTakeFee | Internal 🔒 | | |

| ^L | shouldTakeFeer | 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 ! |

|||||



CONTRACT ASSESMENT

```
| **Auth** | Implementation | |||
| ^ | <Constructor> | Public ! | ● | NO ! |
| ^ | isOwner | Public ! | | NO ! |
| ^ | isdevwallet | Public ! | | NO ! |
| ^ | transferOwnership | Public ! | ● | onlyOwner |
| ^ | renounceOwnership | Public ! | ● | onlyOwner |
|||||
| **IERC165** | Interface | |||
| ^ | supportsInterface | External ! | | NO ! |
|||||
| **IBEP20** | Interface | |||
| ^ | totalSupply | External ! | | NO ! |
| ^ | decimals | External ! | | NO ! |
| ^ | symbol | External ! | | NO ! |
| ^ | name | External ! | | NO ! |
| ^ | getOwner | External ! | | NO ! |
| ^ | balanceOf | External ! | | NO ! |
| ^ | transfer | External ! | ● | NO ! |
| ^ | burning | External ! | ● | NO ! |
| ^ | allowance | External ! | | NO ! |
| ^ | approve | External ! | ● | NO ! |
| ^ | transferFrom | External ! | ● | NO ! |
|||||
| **IDEXFactory** | Interface | |||
| ^ | createPair | External ! | ● | NO ! |
```



CONTRACT ASSESMENT

|||||

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

| [Ⓛ] | factory | External ! | | NO ! |

| [Ⓛ] | WETH | External ! | | NO ! |

| [Ⓛ] | addLiquidity | External ! | ● | NO ! |

| [Ⓛ] | addLiquidityETH | External ! | ⓈⓉ | NO ! |

| [Ⓛ] | swapExactTokensForTokensSupportingFeeOnTransferTokens | External ! | ● | NO ! |

| [Ⓛ] | swapExactETHForTokensSupportingFeeOnTransferTokens | External ! | ⓈⓉ | NO ! |

| [Ⓛ] | swapExactTokensForETHSupportingFeeOnTransferTokens | External ! | ● | NO ! |

|||||

| ****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 🔒 | | |

|||||





CONTRACT ASSESMENT

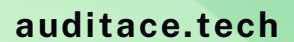
IDividendDistributor	Interface			
^L	setDistributionCriteria	External !		NO !
^L	setShare	External !		NO !
^L	deposit	External !		NO !
^L	process	External !		NO !

| Symbol | Meaning |

| :-----: |-----|

|  | Function can modify state |

|  | Function is payable |



an static analysis of the code were performed using
slither. No issues were found



FUNCTIONAL TESTING

1- Adding liquidity (passed):

<https://testnet.bscscan.com/tx/0xca32ba0387f67d9593fe83462739f4c03a8adeee1be451a7059da0d67274993d>

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

<https://testnet.bscscan.com/tx/0x4ea4d1715fd38dc9e1db7f0f0d7fa73b94d760f3b237ff26aceebb270986bacc>

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

<https://testnet.bscscan.com/tx/0xaa1b997b3a50d9a8a18af69e3dd2e6b2b6c935270da116c5fd88f1fd66c348c3>

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

<https://testnet.bscscan.com/tx/0x5481865f7701295acf9b94ee56a11f0a2e004ab124d259062683fc707343a0ca>

5- Buying when not excluded from fees (up to 25% tax) (passed):

<https://testnet.bscscan.com/tx/0xe072297a293644457422da975fc051c2fbcb9284835200e43a69452e210c68e6>

6- Selling when not excluded from fees (up to 25% tax) (passed):

<https://testnet.bscscan.com/tx/0x6a17b33ff1db8f96c39a33d0eda8bd500a2e9be6d205e9ef63aa6e6ef20b894c>

7- Transferring when not excluded from fees (up to 25% tax) (passed):

<https://testnet.bscscan.com/tx/0x127230cd675cdc11ebe35bed691b2f22d00dd0048ae38642c272715a84916474>



FUNCTIONAL TESTING

7- Rewards, Auto liquidity, Auto Burn, Marketing fee (passed):

<https://testnet.bscscan.com/tx/0x0deea27059d91ac6381abc40efe1c49bbd59efc497e6e7450232f085284ce9a2>

FUNCTIONAL TESTING

Logical – 0 swap threshold can disable trades

Severity: **Critical**

function: setSwapBackSettings

Status: **Not Resolved**

Overview:

if swapThreshold is set to 0, internal swap would be failed due to a division by zero error. (at swapBack function)

```
function setSwapBackSettings(  
    bool _enabled,  
    uint256 _amount  
) external authorized {  
    swapEnabled = _enabled;  
    swapThreshold = _amount;  
}
```

Suggestion

To mitigate this Logical issue, make sure that swapTokensAtAmount is always greater than 0

```
function setSwapBackSettings(  
    bool _enabled,  
    uint256 _amount  
) external authorized {  
    swapEnabled = _enabled;  
    swapThreshold = _amount;  
    require(swapThreshold > totalSupply / 1000000);  
}
```

FUNCTIONAL TESTING

Logical – Overflow at buyback settings

Severity: **Critical**

function: getMultipliedFee & shouldAutoBuyback

Status: **Not Resolved**

Overview:

setBuybackMultiplierSettings and setAutoBuybackSettings functions can be used to set buybackMultiplierLength or autoBuybackBlockPeriod to max uint256 which when added to buybackMultiplierTriggeredAt or autoBuybackBlockLast (if is more than 0) leads to an overflow error resulting in disabled sells for regular wallets. Additionally setting buybackMultiplierLength 0 can cause a division by zero error at getMultipliedFee function.

```
function getMultipliedFee() public view returns (uint256) {
    if (
        buybackMultiplierTriggeredAt.add(buybackMultiplierLength) >
        block.timestamp
    ) {

function setBuybackMultiplierSettings(
    uint256 numerator,
    uint256 denominator,
    uint256 length
) external authorized {
    require(numerator / denominator <= 2 && numerator > denominator);
    buybackMultiplierNumerator = numerator;
    buybackMultiplierDenominator = denominator;
    buybackMultiplierLength = length;
}
```

Suggestion

To mitigate this Logical issue, make sure that **buybackMultiplierLength** or **autoBuybackBlockPeriod** is within a reasonable range (for example 30 minute to 7 weeks)

```
function setBuybackMultiplierSettings(
    uint256 numerator,
    uint256 denominator,
    uint256 length
) external authorized {
    require(numerator / denominator <= 2 && numerator > denominator);
    buybackMultiplierNumerator = numerator;
    buybackMultiplierDenominator = denominator;
    buybackMultiplierLength = length;
    requirie(buubackMultiplierLength <= weeks);
    requirie(buubackMultiplierLength >= 30 minutes);
}
```

FUNCTIONAL TESTING

Centralization – LP tokens going to an EOA

Severity: **Medium**

function: swapBack

Status: **Not Resolved**

Overview:

LP tokens generated from auto-liquidity are sent to an EOA account, this accumulated tokens may be used by a malicious owner to remove a portion of tokens from the pool

```
if (amountToLiquify > 0) {  
    router.addLiquidityETH({value: amountBNBLiquidity}(  
        address(this),  
        amountToLiquify,  
        0,  
        0,  
        autoLiquidityReceiver,  
        block.timestamp  
    ));  
    emit AutoLiquify(amountBNBLiquidity, amountToLiquify);  
}
```

Suggestion

To mitigate this Centralization issue:

- Burn the new LP tokens

or

- Lock the new LP tokens

FUNCTIONAL TESTING

Centralization – Excessive fees

Severity: **Medium**

Status: not resolved

Overview:

Fees can be up to 25 % for each type of tax (buy/sell/transfer). This is 25% more than what is declared in pinksale safu criteria.(50% in total buy+sell)

```
function _setFees(  
    uint256 _liquidityFee,  
    uint256 _buybackFee,  
    uint256 _reflectionFee,  
    uint256 _marketingFee,  
    uint256 _feeDenominator  
) internal {  
    liquidityFee = _liquidityFee;  
    buybackFee = _buybackFee;  
    reflectionFee = _reflectionFee;  
    marketingFee = _marketingFee;  
    totalFee = _liquidityFee.add(_buybackFee).add(_reflectionFee).add(  
        _marketingFee  
    );  
    feeDenominator = _feeDenominator;  
    require(  
        totalFee < feeDenominator / 4,  
        "Total fee should not be greater than 1/4 of fee denominator"  
    );  
}
```

Recommendation:

Ensure that sum of total buy and total sell fees is less than 25%



FUNCTIONAL TESTING

Informational – Use of external token

Status: Not Resolved

Overview:

Contract is using 0xbA2aE424d960c26247Dd6c32edC70B295c744C43 (Dogecoin) as the reward token. This contract is beyond scope of this audit and we assume that this token has enough liquidity and doesn't have major issues that could disable the trades and affect reward system of MILADY token.



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>
