Cyberscope

Audit Report

Metasexxx

April 2022

Type	BEP20
Network	BSC
Address	0xcC017b37D4e719d2cae7B01081638fb635684406
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	Token
Compiler Version	v0.8.6+commit.11564f7e
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xcC017b37D4e719d2cae7B01081638fb635684406
Symbol	MSEX
Decimals	9
Total Supply	1,000,000,000
Domain	metasexxx.app

Source Files

Filename	**SHA256**
contract.sol	7a5d2c1cf11c7252157a4435e06cc670d1076e24d79765ab46fdbbc001ae6334

Audit Updates

Initial Audit	14th April 2022
Corrected	

Contract Analysis

🔴 Critical 🟠 Medium ⚪ Minor 🔵 Pass

Severity	Code	Description
🔵	ST	Contract Owner is not able to stop or pause transactions
🔵	OCTD	Contract Owner is not able to transfer tokens from specific address
🔵	OTUT	Owner Transfer User's Tokens
🟠	ELFM	Contract Owner is not able to increase fees more than a reasonable percent (25%)
🔵	ULTW	Contract Owner is not able to increase the amount of liquidity taken by dev wallet more than a reasonable percent
🔵	MT	Contract Owner is not able to mint new tokens
🔵	BT	Contract Owner is not able to burn tokens from specific wallet
🔵	BC	Contract Owner is not able to blacklist wallets from selling

ELFM - Exceed Limit Fees Manipulation

Criticality	medium
Location	contract.sol#L931

Description

The contract owner has the authority to increase over the allowed limit of 25%. The owner may take advantage of it by calling the `setAllFeePercent` function with a total fee of 50%.

```
function setAllFeePercent(uint8 taxFee, uint8 liquidityFee, uint8 burnFee, uint8 
walletFee, uint8 buybackFee) external onlyOwner() {
    require(taxFee >= 0 && taxFee <=maxTaxFee,"TF err");
    require(liquidityFee >= 0 && liquidityFee <=maxLiqFee,"LF err");
    require(burnFee >= 0 && burnFee <=maxBurnFee,"BF err");
    require(walletFee >= 0 && walletFee <=maxWalletFee,"WF err");
    require(buybackFee >= 0 && buybackFee <=maxBuybackFee,"BBF err");
    _taxFee = taxFee;
    _liquidityFee = liquidityFee;
    _burnFee = burnFee;
    _buybackFee = buybackFee;
    _walletFee = walletFee;
}
```

Recommendation

The contract could embody a check for the maximum acceptable value.

The team should carefully manage the private keys of the owner’s account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. That risk can be prevented by temporarily locking the contract or renouncing ownership.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L07	Missing Events Arithmetic
●	L08	Tautology or Contradiction
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L499,508,514,519,527,815,819,823,827,836,841,845,850,856,861,866,870,874,883,900,923,927,944,966,1067,1324

Description

Public functions that are never called by the contract should be declared external to save gas.

```
recoverBEP20
isExcludedFromFee
setSwapAndLiquifyEnabled
buyBackUpperLimitAmount
includeInFee
excludeFromFee
excludeFromReward
reflectionFromToken
deliver
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L702,706,708,704,705,707,709,710,729,721

Description

Constant state variables should be declared constant to save gas.

```
router
mintedByMudra
minMxWalletPercentage
minMxTxPercentage
maxWalletFee
maxTaxFee
maxLiqFee
maxBuybackFee
maxBurnFee
...
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L559,966,1031,1037,725,731,732,735,738,741,744,747,757,758

Description

Solidity defines a naming convention that should be followed. Rule exceptions:

- Allow constant variable name/symbol/decimals to be lowercase.
- Allow _ at the beginning of the mixed_case match for private variables and unused parameters.

```
_maxWalletAmount
_maxTxAmount
_buybackFee
_walletFee
_burnFee
_liquidityFee
_taxFee
_symbol
_name
...
```

Recommendation

Follow the Solidity naming convention.
https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L931,948,952,959

Description

Detected missing events for critical arithmetic parameters. There are functions that have no event emitted, so it is difficult to track off-chain changes.

```
_maxWalletAmount = _tTotal.mul(maxWalletPercent).div(10 ** 2)
_maxTxAmount = _tTotal.mul(maxTxPercent).div(10 ** 2)
buyBackUpperLimit = buyBackLimit * 10 ** 18
_taxFee = taxFee
```

Recommendation

Emit an event for critical parameter changes.

L08 - Tautology or Contradiction

Criticality	minor
Location	contract.sol#L931

Description

Detects expressions that are tautologies or contradictions. For instance, an uint variable will always be greater than or equal to zero.

```
require(bool,string)(burnFee >= 0 && burnFee <= maxBurnFee,BF err)
require(bool,string)(liquidityFee >= 0 && liquidityFee <= maxLiqFee,LF err)
require(bool,string)(buybackFee >= 0 && buybackFee <= maxBuybackFee,BBF err)
require(bool,string)(taxFee >= 0 && taxFee <= maxTaxFee,TF err)
require(bool,string)(walletFee >= 0 && walletFee <= maxWalletFee,WF err)
```

Recommendation

Fix the incorrect comparison by changing the value type or the comparison.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L358,318,328,343,353,265,292,436,409,425,420,394,398

Description

Functions that are not used in the contract, and make the code's size bigger.

```
safeTransferFrom
safeTransfer
safeIncreaseAllowance
safeDecreaseAllowance
safeApprove
_callOptionalReturn
sendValue
isContract
functionCallWithValue
...
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L1142

Description

Performing divisions before multiplications may cause lose of prediction.

```
spentAmount = contractTokenBalance.div(totFee).mul(_buybackFee)
spentAmount = contractTokenBalance.div(totFee).mul(_walletFee)
spentAmount = contractTokenBalance.div(totFee).mul(_burnFee)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
Address	Library			
	isContract	Internal		
	sendValue	Internal	✓	
	functionCall	Internal	✓	
	functionCall	Internal	✓	
	functionCallWithValue	Internal	✓	
	functionCallWithValue	Internal	✓	

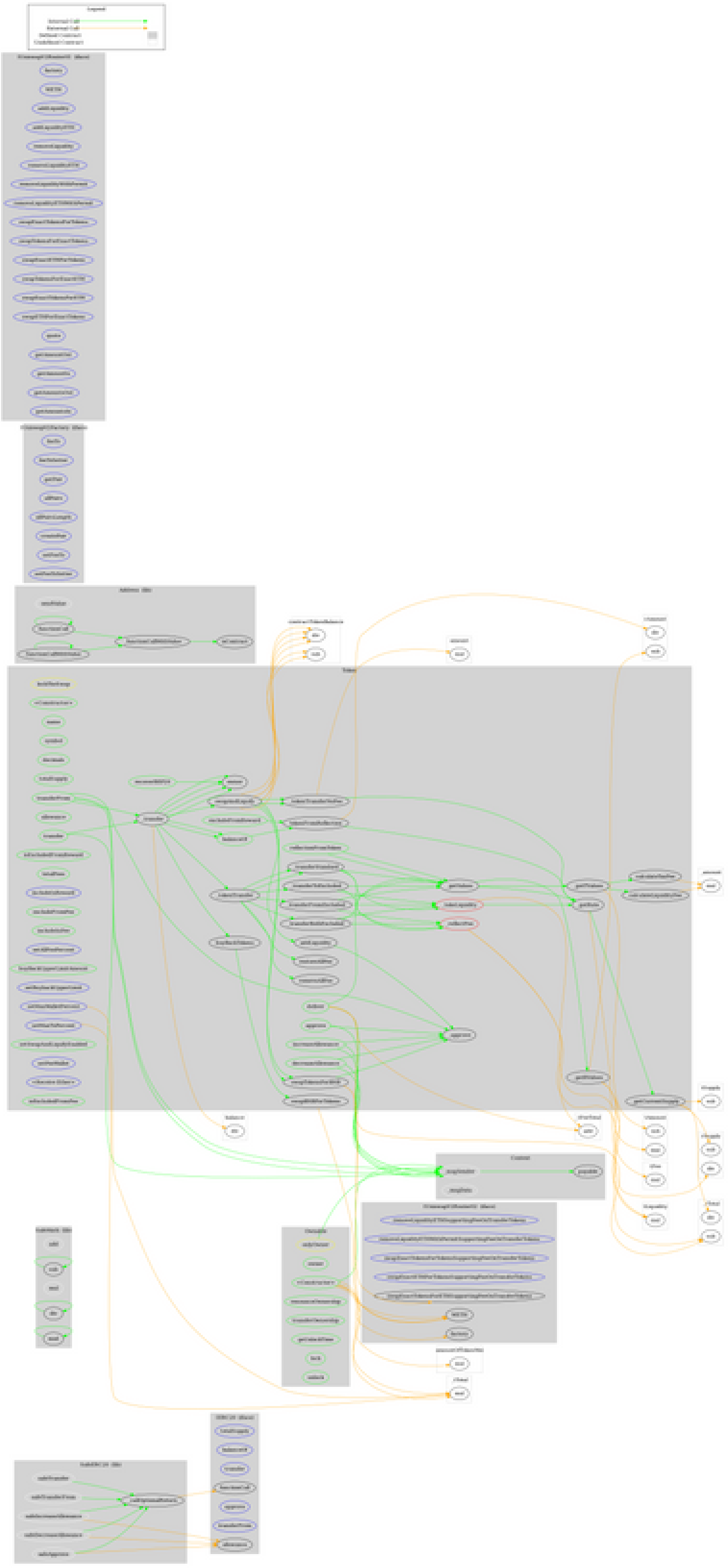
	_functionCallWithValue	Private	✓	
SafeERC20	Library			
	safeTransfer	Internal	✓	
	safeTransferFrom	Internal	✓	
	safeApprove	Internal	✓	
	safeIncreaseAllowance	Internal	✓	
	safeDecreaseAllowance	Internal	✓	
	_callOptionalReturn	Private	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	geUnlockTime	Public		-
	lock	Public	✓	onlyOwner
	unlock	Public	✓	-
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-

	removeLiquidity	External	✓	-
	removeLiquidityETH	External	✓	-
	removeLiquidityWithPermit	External	✓	-
	removeLiquidityETHWithPermit	External	✓	-
	swapExactTokensForTokens	External	✓	-
	swapTokensForExactTokens	External	✓	-
	swapExactETHForTokens	External	Payable	-
	swapTokensForExactETH	External	✓	-
	swapExactTokensForETH	External	✓	-
	swapETHForExactTokens	External	Payable	-
	quote	External		-
	getAmountOut	External		-
	getAmountIn	External		-
	getAmountsOut	External		-
	getAmountsIn	External		-
IUniswapV2Router02	Interface	IUniswapV2Router01		
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
Token	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-

	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	deliver	Public	✓	-
	reflectionFromToken	Public		-
	tokenFromReflection	Public		-
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setAllFeePercent	External	✓	onlyOwner
	buyBackUpperLimitAmount	Public		-
	setBuybackUpperLimit	External	✓	onlyOwner
	setMaxTxPercent	External	✓	onlyOwner
	setMaxWalletPercent	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	setFeeWallet	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_reflectFee	Private	✓	
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-

	_approve	Private	✓	
	_transfer	Private	✓	
	swapAndLiquify	Private	✓	lockTheSwap
	buyBackTokens	Private	✓	lockTheSwap
	swapTokensForBNB	Private	✓	
	swapBNBForTokens	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	
	_tokenTransferNoFee	Private	✓	
	recoverBEP20	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	metasexxx.app
Registry Domain ID	4839A4FDC-APP
Creation Date	2022-01-10T19:54:54Z
Updated Date	2022-01-15T19:54:54Z
Registry Expiry Date	2023-01-10T19:54:54Z
Registrar WHOIS Server	whois.google.com
Registrar URL	domains.google
Registrar	Google LLC.
Registrar IANA ID	895

The domain has been created 3 months before the creation of the audit. It will expire in 9 months.

There is no public billing information, the creator is protected by the privacy settings.

Summary

Metasexxx is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler error or critical issues. The contract Owner can access some admin functions that can not be used in a malicious way to disturb the users' transactions. There is a limit of max 50% fees.

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.

Cyberscope 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 Cyberscope 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 Cyberscope 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 Cyberscope team disclaims any liability for the resulting losses.

About Cyberscope

Coinscope audit and K.Y.C. service has been rebranded to Cyberscope.

Coinscope is the leading early coin listing, voting and auditing authority firm. The audit process is analyzing and monitoring many aspects of the project. That way, it gives the community a good sense of security using an informative report and a generic score.

Cyberscope and Coinscope are aiming to make crypto discoverable and efficient globally. They provides all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io