



Cyberscope

Audit Report

Madrox Finance

May 2022

Type BEP20

Network BSC

Address 0x2ceE4aF3439E01Bc8De4908eE04D1989E8041f20

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	3
Contract Analysis	4
Contract Diagnostics	5
MTS - Manipulate Total Supply	6
Description	6
Recommendation	6
L01 - Public Function could be Declared External	7
Description	7
Recommendation	7
L02 - State Variables could be Declared Constant	8
Description	8
Recommendation	8
L04 - Conformance to Solidity Naming Conventions	9
Description	9
Recommendation	9
L05 - Unused State Variable	10
Description	10
Recommendation	10
L09 - Dead Code Elimination	11
Description	11
Recommendation	11
L13 - Divide before Multiply Operation	12
Description	12

Recommendation	12
Contract Functions	13
Contract Flow	18
Domain Info	19
Summary	20
Disclaimer	21
About Cyberscope	22

Contract Review

Contract Name	MadroxFinance
Compiler Version	v0.7.4+commit.3f05b770
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x2ceE4aF3439E01Bc8De4908eE04D1989E8041f20
Symbol	MDR
Decimals	5
Total Supply	10,178,776
Domain	madrox.finance

Source Files

Filename	SHA256
contract.sol	74c715bec2d7ded6bf133df292004a5669cce9475fdfce41cdba337d5f59b285

Audit Updates

Initial Audit	8th May 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

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	MTS	Manipulate Total Supply
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation

MTS - Manipulate Total Supply

Criticality	minor
Location	contract.sol#L527

Description

Owner is able to manipulate total supply. This change will have a direct impact on the token price and Market Cap

```
_totalSupply = tmp;  
_gonsPerFragment = TOTAL_GONS.div(tmp);  
_lastRebasedTime = block.number;  
  
pairContract.sync();
```

Recommendation

The contract owner should carefully manage the adjustment of the circulating supply (increases or decreases), according to the token's price fluctuations.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L357,362,388,392,396,913,932

Description

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

```
setPairAddress  
getLiquidityBacking  
decimals  
symbol  
name  
transferOwnership  
renounceOwnership
```

Recommendation

Use the external attribute for functions never called from the contract.

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L435,436,410,408,409,433,444

Description

Constant state variables should be declared constant to save gas.

```
swapEnabled  
feeDenominator  
_symbol  
_name  
_decimals  
ZERO  
DEAD
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L146,147,164,184,811,820,883,904,905,906,923,927,932,936,408,409,410,413,435,436,458,459,462,463,464,465

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.

```
_totalSupply  
_lastAddLiquidityTime  
_lastRebasedTime  
_initRebaseStartTime  
_autoAddLiquidity  
_autoRebase  
ZERO  
DEAD  
_isFeeExempt  
...
```

Recommendation

Follow the Solidity naming convention.

<https://docs.soliditylang.org/en/v0.4.25/style-guide.html#naming-conventions>

L05 - Unused State Variable

Criticality

minor

Location

contract.sol#L13

Description

There are segments that contain unused state variables.

```
MAX_INT256
```

Recommendation

Remove unused state variables.

L09 - Dead Code Elimination

Criticality

minor

Location

contract.sol#L41

Description

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

```
abs
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality minor

Location contract.sol#L527,645,913

Description

Performing divisions before multiplications may cause lose of prediction.

```
liquidityBalance = _gonBalances[pair].div(_gonsPerFragment)
_gonBalances[lpReceiver] =
_gonBalances[lpReceiver].add(gonAmount.div(feeDenominator).mul(liquidityFee))
_gonBalances[address(this)] =
_gonBalances[address(this)].add(gonAmount.div(feeDenominator).mul(maintainFee))
_gonBalances[firePit] =
_gonBalances[firePit].add(gonAmount.div(feeDenominator).mul(burnFee))
feeAmount = gonAmount.div(feeDenominator).mul(_totalFee)
tmp = tmp.mul(deno.mul(100).add(hourRebaseRate)).div(deno.mul(100))
tmp = tmp.mul(deno.mul(100).add(minuteRebaseRate)).div(deno.mul(100))
tmp = tmp.mul(deno.mul(100).add(rebaseRate)).div(deno.mul(100))
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMathInt	Library			
	mul	Internal		
	div	Internal		
	sub	Internal		
	add	Internal		
	abs	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	transfer	External	✓	-
	approve	External	✓	-
	transferFrom	External	✓	-
IPancakeSwap Pair	Interface			
	name	External		-
	symbol	External		-
	decimals	External		-

	totalSupply	External		-
	balanceOf	External		-
	allowance	External		-
	approve	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	mint	External	✓	-
	burn	External	✓	-
	swap	External	✓	-
	skim	External	✓	-
	sync	External	✓	-
	initialize	External	✓	-
IPancakeSwap Router	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		-
	removeLiquidityETHSupportingFeeOnTransferTokens	External	✓	-
	removeLiquidityETHWithPermitSupportingFeeOnTransferTokens	External	✓	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IPancakeSwapFactory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
Ownable	Implementation			
	<Constructor>	Public	✓	-
	owner	Public		-
	isOwner	Public		-
	renounceOwnership	Public	✓	onlyOwner

	transferOwnership	Public	✓	onlyOwner
	_transferOwnership	Internal	✓	
ERC20Detailed	Implementation	IERC20		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
MadroxFinance	Implementation	ERC20Detailed, Ownable		
	<Constructor>	Public	✓	ERC20Detailed Ownable
	setZeroFees	External	✓	onlyOwner
	setNormalFees	External	✓	onlyOwner
	manualRebase	External	✓	onlyOwner
	rebase	Internal	✓	
	transfer	External	✓	validRecipient
	transferFrom	External	✓	validRecipient
	_basicTransfer	Internal	✓	
	_transferFrom	Internal	✓	
	takeFee	Internal	✓	
	addLiquidity	Internal	✓	swapping
	swapBack	Internal	✓	swapping
	withdrawAllToTreasury	External	✓	swapping onlyOwner
	shouldTakeFee	Internal		
	shouldRebase	Internal		
	shouldAddLiquidity	Internal		
	shouldSwapBack	Internal		
	setAutoRebase	External	✓	onlyOwner
	setAutoAddLiquidity	External	✓	onlyOwner
	allowance	External		-
	decreaseAllowance	External	✓	-
	increaseAllowance	External	✓	-
	approve	External	✓	-

	checkFeeExempt	External		-
	getCirculatingSupply	Public		-
	isNotInSwap	External		-
	manualSync	External	✓	-
	setFeeReceivers	External	✓	onlyOwner
	getLiquidityBacking	Public		-
	setWhitelist	External	✓	onlyOwner
	setBotBlacklist	External	✓	onlyOwner
	setPairAddress	Public	✓	onlyOwner
	setLP	External	✓	onlyOwner
	totalSupply	External		-
	balanceOf	External		-
	isContract	Internal		
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	madrox.finance
Registry Domain ID	def605d538a845a897fa1d285f855a6f-DONUTS
Creation Date	2022-04-16T16:54:56Z
Updated Date	2022-04-21T16:55:10Z
Registry Expiry Date	2023-04-16T16:54:56Z
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	https://www.namecheap.com/
Registrar	NameCheap, Inc.
Registrar IANA ID	1068

The domain has been created 22 days before the creation of the audit. It will expire in 11 months.

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

Summary

Madrox Finance 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.

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>