Cyberscope

Audit Report

MADCHAD TOKEN

October 2022

Type	BEP20
Network	BSC
Address	0x1F91F6c93023A441c3a72B24120E19acEF677911
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	MADCHADTOKEN
Compiler Version	v0.8.17+commit.8df45f5f
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0x1F91F6c93023A441c3a72B24120E19acEF677911
Symbol	MADCHAD
Decimals	18
Total Supply	250,000,000
Domain	https://madchad.io

Source Files

Filename	**SHA256**
contract.sol	fc788ac9d379ce29b0533548b824263e175b4551cc32decdeb26519610917e27

Audit Updates

Initial Audit	25th October 2022
Corrected	

Contract Analysis

● Critical ● Medium ● Minor / Informative ● Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OCTD	Transfers Contract's Tokens	Unresolved
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	ULTW	Transfers Liquidity to Team Wallet	Unresolved
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

OCTD - Transfers Contract's Tokens

Criticality	minor / informative
Location	contract.sol#L781
Status	Unresolved

Description

The contract owner has the authority to claim all the balance of the contract. The owner may take advantage of it by calling the `claimTokens` function.

```
function rescueBSC20(address tokenAdd, uint256 amount) external onlyOwner {
        IERC20(tokenAdd).transfer(devWallet, amount);
}
```

Recommendation

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.

ULTW - Transfers Liquidity to Team Wallet

Criticality	minor / informative
Location	contract.sol#L777
Status	Unresolved

Description

The contract owner has the authority to transfer funds without limit to the team wallet. These funds have been accumulated from fees collected from the contract. The owner may take advantage of it by calling the `method1` and `method2` methods.

```
function rescueBNB(uint256 weiAmount) external onlyOwner {
        payable(devWallet).transfer(weiAmount);
}
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may volatile the token's price.

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 / Informative

Severity	Code	Description	Status
	STC	Succeeded Transfer Check	Unresolved
	CO	Code Optimization	Unresolved
	L02	State Variables could be Declared Constant	Unresolved
	L04	Conformance to Solidity Naming Conventions	Unresolved
	L05	Unused State Variable	Unresolved
	L07	Missing Events Arithmetic	Unresolved
	L09	Dead Code Elimination	Unresolved
	L13	Divide before Multiply Operation	Unresolved
	L14	Uninitialized Variables in Local Scope	Unresolved

STC - Succeeded Transfer Check

Criticality	minor / informative
Location	contract.sol#L782
Status	Unresolved

Description

According to the ERC20 specification, the transfer methods should be checked if the result is successful. Otherwise, the contract may wrongly assume that the transfer has been established.

```
IERC20(tokenAdd).transfer(devWallet, amount);
```

Recommendation

The contract should check if the result of the transfer methods is successful.

CO - Code Optimization

Criticality	minor / informative
Location	contract.sol#L552
Status	Unresolved

Description

There are code segments that could be optimized. A segment may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer operations. The variables `feeswap` and `feesum` are set with the same value. As a result one of them is redundant.

```
        else if (recipient == pair && !useLaunchFee) {
            feeswap = sellTaxes.marketing + sellTaxes.gameevolution + sellTaxes.staking +
sellTaxes.liquidity + sellTaxes.dev;
            feesum = feeswap;
            currentTaxes = sellTaxes;
            _taxDenominator = sellTaxes.denominator;
        } else if (!useLaunchFee) {
            feeswap = taxes.marketing + taxes.gameevolution + taxes.staking + taxes.liquidity +
taxes.dev;
            feesum = feeswap;
            currentTaxes = taxes;
            _taxDenominator = taxes.denominator;
        } else if (useLaunchFee) {
            feeswap = launchtax;
            feesum = launchtax;
        }
```

Recommendation

Rewrite some code segments so the runtime will be more performant. It is recommended to remove one of the two variables.

L02 - State Variables could be Declared Constant

Criticality	minor / informative
Location	contract.sol#L453
Status	Unresolved

Description

Constant state variables should be declared constant to save gas.

```
launchtax
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor / informative
Location	contract.sol#L727,459,710,709,451,725,726,767,711,57,712,708,728,703,55,412,611,745,724,723,713
Status	Unresolved

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.

```
_liquidity
deadWallet
_gameevolution
_marketing
genesis_block
_staking
_address
_allowances
UpdateBuyTaxes
...
```

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 / informative
Location	contract.sol#L476
Status	Unresolved

Description

There are segments that contain unused state variables.

```
_lastSell
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor / informative
Location	contract.sol#L703,745
Status	Unresolved

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.

```
tokenLiquidityThreshold = new_amount * 10 ** decimals()
deadline = _deadline
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor / informative
Location	contract.sol#L303
Status	Unresolved

Description

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

```
_burn
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor / informative
Location	contract.sol#L611
Status	Unresolved

Description

Performing divisions before multiplications may cause lose of prediction.

```
unitBalance = deltaBalance / ((denominator - swapTaxes.liquidity) / swapTaxes.denominator)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor / informative
Location	contract.sol#L563,564,567
Status	Unresolved

Description

The are variables that are defined in the local scope and are not initialized.

```
feeswap
feesum
currentTaxes
```

Recommendation

All the local scoped variables should be initialized.

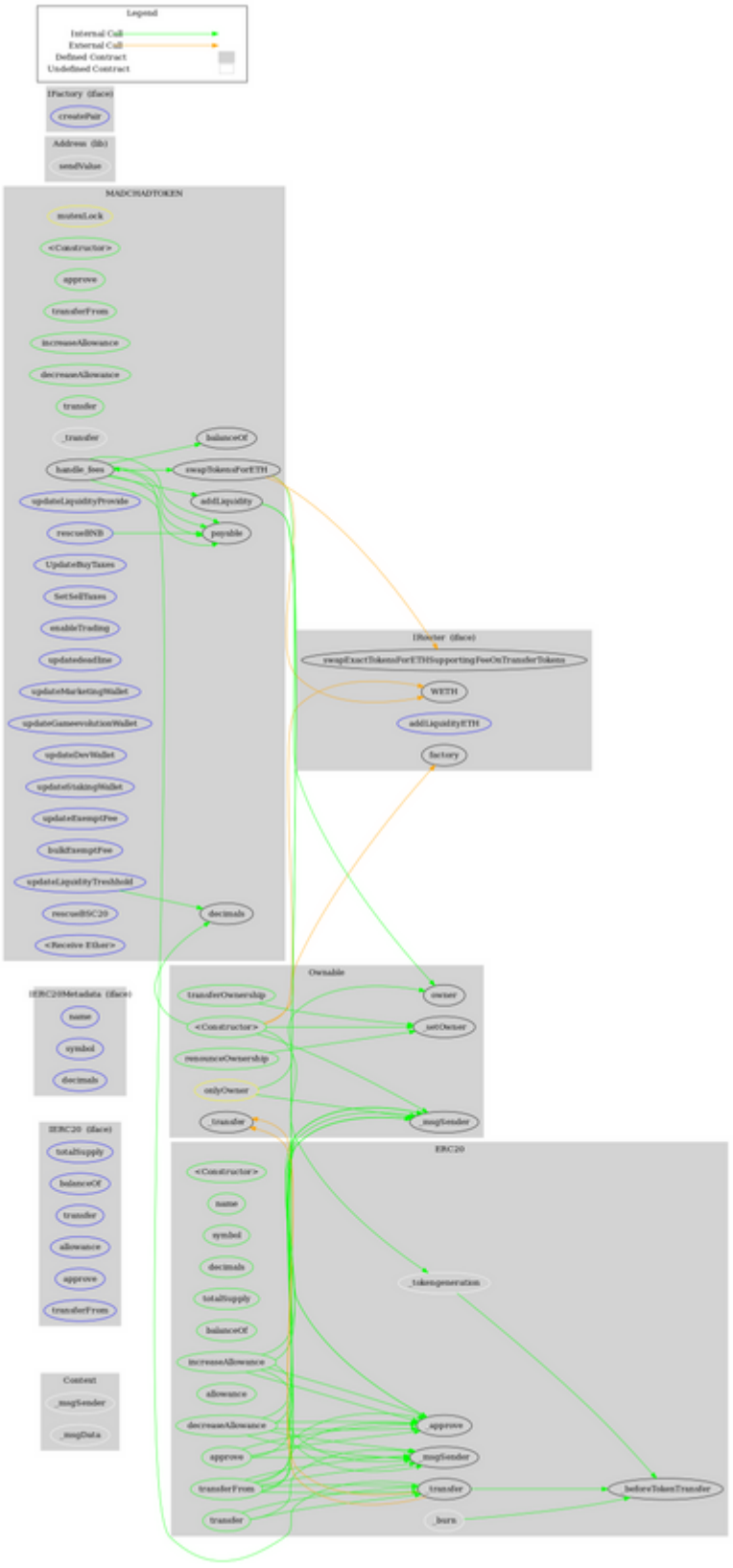
Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-

	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_tokengeneration	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
Address	Library			
	sendValue	Internal	✓	
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	_setOwner	Private	✓	
IFactory	Interface			
	createPair	External	✓	-
IRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
MADCHADTOKEN	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	approve	Public	✓	-
	transferFrom	Public	✓	-

	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	transfer	Public	✓	-
	_transfer	Internal	✓	
	handle_fees	Private	✓	mutexLock
	swapTokensForETH	Private	✓	
	addLiquidity	Private	✓	
	updateLiquidityProvide	External	✓	onlyOwner
	updateLiquidityTreshhold	External	✓	onlyOwner
	UpdateBuyTaxes	External	✓	onlyOwner
	SetSellTaxes	External	✓	onlyOwner
	enableTrading	External	✓	onlyOwner
	updatedeadline	External	✓	onlyOwner
	updateMarketingWallet	External	✓	onlyOwner
	updateGameevolutionWallet	External	✓	onlyOwner
	updateDevWallet	External	✓	onlyOwner
	updateStakingWallet	External	✓	onlyOwner
	updateExemptFee	External	✓	onlyOwner
	bulkExemptFee	External	✓	onlyOwner
	rescueBNB	External	✓	onlyOwner
	rescueBSC20	External	✓	onlyOwner
	<Receive Ether>	External	Payable	-

Contract Flow



Domain Info

Domain Name	madchad.io
Registry Domain ID	252ff5923d2f4c19911a262cb42419c0-DONUTS
Creation Date	2022-09-10T13:53:39Z
Updated Date	2022-10-01T14:54:23Z
Registry Expiry Date	2023-09-10T13:53:39Z
Registrar WHOIS Server	whois.rrpproxy.net
Registrar URL	http://key-systems.net
Registrar	Key-Systems GmbH
Registrar IANA ID	269

The domain was created about 1 month 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

Madchad 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 also a limit of max 10% 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 provide all the essential tools to assist users draw their own conclusions.

The Cyberscope team

https://www.cyberscope.io