Cyberscope

Audit Report

MarsDoge

April 2022

Type	BEP20
Network	BSC
Address	0x0972687E5aC075C53690A0524a099D22e24a9438
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	MarsDoge
Compiler Version	v0.8.7+commit.e28d00a7
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x0972687E5aC075C53690A0524a099D22e24a9438
Symbol	MarsDoge
Decimals	6
Total Supply	1,000,000,000,000,000
Domain	marsdoge.co

Source Files

Filename	**SHA256**
contract.sol	3757751e8b3a72b31465dd6246262393601f382bbb8e5d45a68a98e8e4a9908f

Audit Updates

Initial Audit	21st 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

ULTW - Unlimited Liquidity to Team Wallet

Criticality	minor
Location	contract.sol#L877

Description

The contract owner has the authority to transfer funds to the team wallet. These funds have been accumulated from fees collected from the contract. The contract is using 30% of the funds for the liquidity pool and send all the remaining to the dev's wallet.

```
function swapAndLiquify(uint256 canSwapAmount) public {
    if(balanceOf(address(this)) >= canSwapAmount){
        uint256 half32 = canSwapAmount.div(3).mul(2);
        uint256 otherHalf = canSwapAmount.sub(half32);
        uint256 initialBalance = address(this).balance;
        swapTokensForEth(half32);
        uint256 newBalance = address(this).balance.sub(initialBalance);
        addLiquidity(otherHalf, newBalance.div(2));
    }
    uint256 haveBnb = address(this).balance;
    if(haveBnb > 10**16){
        _receive.transfer(haveBnb);
    }
}
```

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

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
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation
●	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L414,423,429,703,707,711,715,724,729,733,738,743,749,754,759,763,773,777,781,791,818

Description

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

```
uniswapV2PairSync
setSwapAndLiquifyEnabled
includeInFee
changeswapAction
excludeFromFee
totalFees
isExcludedFromReward
decreaseAllowance
increaseAllowance
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L668,665,669,666

Description

Constant state variables should be declared constant to save gas.

```
_symbol
_receive
_name
_destroyAddress
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L395,470,471,488,510,791,663,669,670

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.

```
_tokenOwner
_receive
_maxTxAmount
_enabled
WETH
MINIMUM_LIQUIDITY
PERMIT_TYPEHASH
DOMAIN_SEPARATOR
_owner
```

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#L785

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.

```
_maxTxAmount = _tTotal.mul(maxTxPercent).div(10 ** 4)
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L358,318,328,343,353,265,292,798,823

Description

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

```
_takeLiquidity
_getRValues
sendValue
isContract
functionCallWithValue
functionCall
_functionCallWithValue
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L877,917

Description

Performing divisions before multiplications may cause lose of prediction.

```
Transfer(sender,recipient,tAmount.div(100).mul(recipientRate))
_rOwned[recipient] = _rOwned[recipient].add(rAmount.div(100).mul(recipientRate))
_takeTransfer(sender,address(this),tAmount.div(100).mul(6),currentRate)
half32 = canSwapAmount.div(3).mul(2)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L923

Description

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

```
rate
```

Recommendation

All the local scoped variables should be initialized.

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	✓	

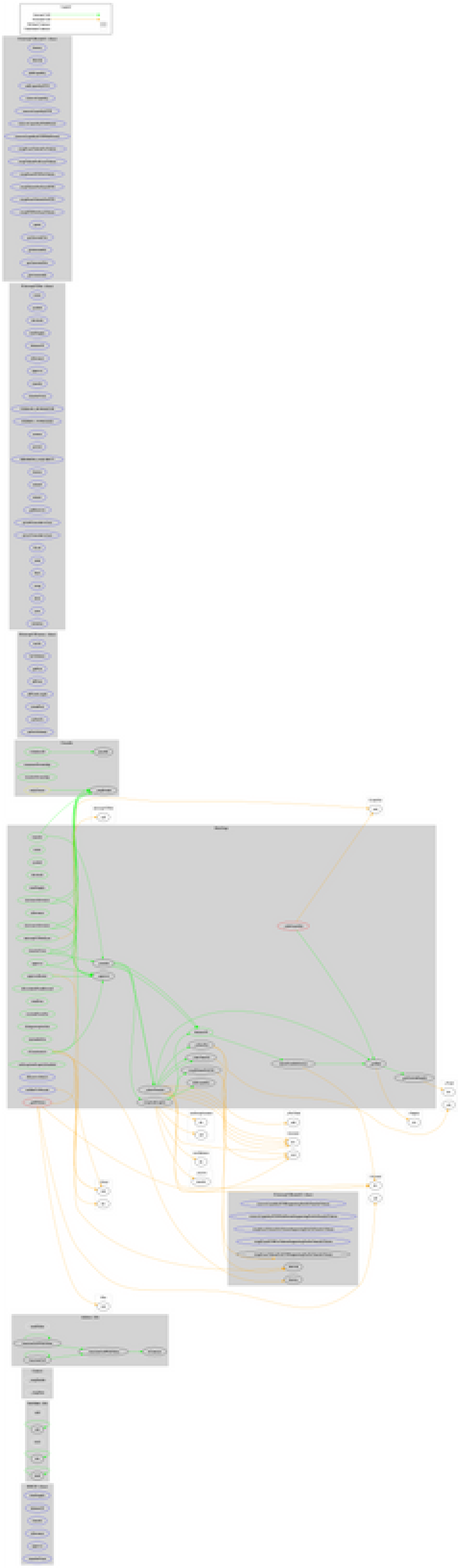
	_functionCallWithValue	Private	✓	
Ownable	Implementation	Context		
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
	whitdrawB	Public	✓	onlyOwner
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	createPair	External	✓	-
	setFeeTo	External	✓	-
	setFeeToSetter	External	✓	-
IUniswapV2Pair	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	✓	-
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	✓	-
MarsDoge	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	approveRouter	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	isExcludedFromReward	Public		-
	totalFees	Public		-
	tokenFromReflection	Public		-
	excludeFromFee	Public	✓	onlyOwner
	changeswapAction	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner
	setMaxTxPercent	External	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_getRValues	Private		
	_getRate	Private		

	_getCurrentSupply	Private		
	uniswapV2PairSync	Public	✓	-
	_takeLiquidity	Private	✓	
	_approve	Private	✓	
	_transfer	Private	✓	
	swapAndLiquify	Public	✓	-
	addLiquidity	Private	✓	
	swapTokensForEth	Private	✓	
	_tokenTransfer	Private	✓	
	_takeTransfer	Private	✓	
	_reflectFee	Private	✓	

Contract Flow



Domain Info

Domain Name	marsdoge.co
Registry Domain ID	D6C467E340AD249CBB7E87C181B861493-GDREG
Creation Date	2022-04-12T12:33:11Z
Updated Date	2022-04-17T12:33:11Z
Registry Expiry Date	2023-04-12T12:33:11Z
Registrar WHOIS Server	
Registrar URL	www.wildwestdomains.com
Registrar	Wild West Domains, Inc.
Registrar IANA ID	440

The domain has been created 9 days before the creation of the audit. It will expire in 12 months.

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

Summary

MarsDoge Token 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. The contract sends a lot of funds to the dev's wallet. There is also a fixed limit of 8% 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