Cyberscope

Audit Report

Martian Doge

April 2022

Type	BEP20
Network	BSC
Address	0x0Ea9790b1385ce5efAfc273623A013D007e50254
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	MartianDoge
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0x0Ea9790b1385ce5efAfc273623A013D007e50254
Symbol	Martian Doge
Decimals	9
Total Supply	100,000,000
Domain	martiandoge.top

Source Files

Filename	**SHA256**
contract.sol	b13898e1f5a883d1008717ed4b11acfd4b6ea9b9de8d0451fa48cf5e72d96735

Audit Updates

Initial Audit	6th 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	minor
Location	contract.sol#L997,1002,1007

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 set functions with 10% value and have a total fee of 30%.

```
function setTaxFee(uint256 value) public onlyOwner {
  require(value <= 10, "No crazy fee.");
  _taxFee = value;
}

function setLiquidityFee(uint256 value) public onlyOwner {
  require(value <= 10, "No crazy fee.");
  _liquidityFee = value;
}

function setBurnFee(uint256 value) public onlyOwner {
  require(value <= 10, "No crazy fee.");
  _burnFee = value;
}
```

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.

ULTW - Unlimited Liquidity to Team Wallet

Criticality	medium
Location	contract.sol#L1016

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 setting a high fee to the `_marketingRate` variable.

```
function setMarketingPerc(uint256 value) public onlyOwner {
  _marketingRate = value;
}
```

```
uint256 marketingToken = contractTokenBalance.mul(_marketingRate).div(100);
     //marketing
     swapAndMarketing(marketingToken);
```

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.

BC - Blacklisted Contracts

Criticality	medium
Location	contract.sol#L1028

Description

The contract owner has the authority to stop contracts from transactions. The owner may take advantage of it by calling the `setBlacklist` function.

```
function setBlacklist(address account, bool value) public onlyOwner() {
  isBlacklist[account] = value;
}
```

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.

Contract Diagnostics

● Critical ● Medium ● Minor

Severity	Code	Description
●	CO	Code Optimization
●	CR	Code Repetition
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L05	Unused State Variable
●	L07	Missing Events Arithmetic
●	L09	Dead Code Elimination
●	L13	Divide before Multiply Operation
●	L14	Uninitialized Variables in Local Scope

CO - Code Optimization

Criticality	minor
Location	contract.sol#L1042,1045

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. Both swapAndMarketing and swapAndLiquify will call internally the swapTokensForEth function which is responsible for selling tokens for BNB. This segment could be optimised and converted into 1 call instead, and then split the amount accordingly.

```
    contractTokenBalance = numTokensSellToAddToLiquidity;
      uint256 marketingToken =
contractTokenBalance.mul(_marketingRate).div(100);
      //marketing
      swapAndMarketing(marketingToken);
      contractTokenBalance = contractTokenBalance.sub(marketingToken);
      //add liquidity
      swapAndLiquify(contractTokenBalance);
```

Recommendation

Move swapTokensForEth function to parent level and execute once.

CR - Code Repetition

Criticality	minor
Location	contract.sol#L837,1016

Description

There are code segments that are repetitive in the contract. Those segments increase the code size of the contract unnecessarily. Both functions have the same functionality hence one of them is redundant.

```
function setMarketingRate(uint256 value) public onlyOwner() {
    _marketingRate = value;
  }

  function setMarketingPerc(uint256 value) public onlyOwner {
    _marketingRate = value;
  }
```

Recommendation

Create an internal function that contains the code segment and remove it from all the sections.

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L404,423,432,755,759,763,767,776,781,785 and 21 more

Description

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

```
setMarketingPerc
setBurnFee
setLiquidityFee
setTaxFee
isExcludedFromFee
claimTokens
setSwapAndLiquifyEnabled
setNumTokensSellToAddToLiquidity
includeInFee
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L689,396

Description

Constant state variables should be declared constant to save gas.

```
_previousOwner
deadWallet
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L395,475,476,493,515,879,955,961,675,678 and 2 more

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.

```
_marketingRate
_burnFee
_liquidityFee
_taxFee
_amount
_enabled
WETH
MINIMUM_LIQUIDITY
PERMIT_TYPEHASH
...
```

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#L396

Description

There are segments that contain unused state variables.

```
_previousOwner
```

Recommendation

Remove unused state variables.

L07 - Missing Events Arithmetic

Criticality	minor
Location	contract.sol#L833,875,997,1002,1007,1012

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.

```
_marketingRate = value
_burnFee = value
_liquidityFee = value
_taxFee = value
numTokensSellToAddToLiquidity = swapNumber * 10 ** _decimals
```

Recommendation

Emit an event for critical parameter changes.

L09 - Dead Code Elimination

Criticality	minor
Location	contract.sol#L358,318,328,343,353,265,292,961,955

Description

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

```
calculateTaxFee
calculateLiquidityFee
sendValue
isContract
functionCallWithValue
functionCall
_functionCallWithValue
```

Recommendation

Remove unused functions.

L13 - Divide before Multiply Operation

Criticality	minor
Location	contract.sol#L907,926,938

Description

Performing divisions before multiplications may cause lose of prediction.

```
tBurn = tAmount.mul(_burnFee).div(100)
tLiquidity = tAmount.mul(_liquidityFee).div(100)
tFee = tAmount.mul(_taxFee).div(100)
```

Recommendation

The multiplications should be prior to the divisions.

L14 - Uninitialized Variables in Local Scope

Criticality	minor
Location	contract.sol#L1205,1143,1184,1163

Description

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

```
tFeeAmount
```

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	✓	

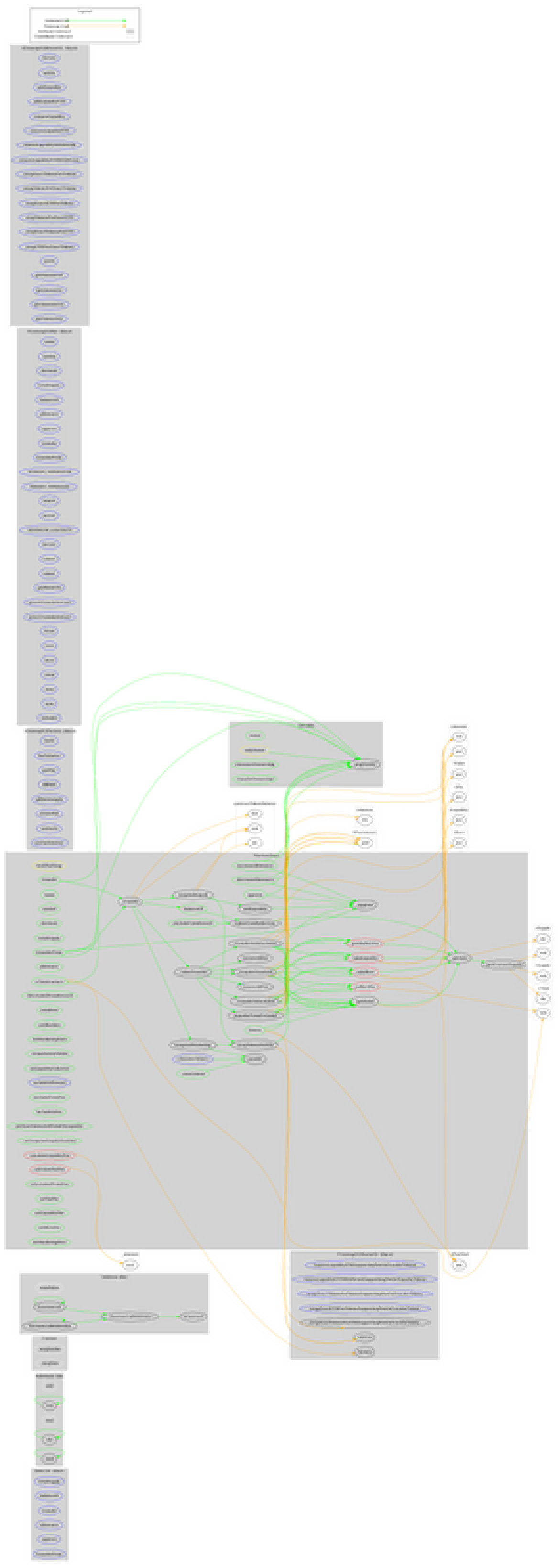
	_functionCallWithValue	Private	✓	
Ownable	Implementation	Context		
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
IUniswapV2Factory	Interface			
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	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	✓	-
MartianDoge	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	✓	-
	tokenFromReflection	Public		-
	setBlacklist	Public	✓	onlyOwner
	setMarketingRate	Public	✓	onlyOwner
	setmarketingWallet	Public	✓	onlyOwner
	setLiquidityCollector	Public	✓	onlyOwner
	excludeFromReward	Public	✓	onlyOwner
	includeInReward	External	✓	onlyOwner
	excludeFromFee	Public	✓	onlyOwner
	includeInFee	Public	✓	onlyOwner

	setNumTokensSellToAddToLiquidity	Public	✓	onlyOwner
	setSwapAndLiquifyEnabled	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_reflectFee	Private	✓	
	_getRate	Private		
	_getRated	Private		
	_getReflectFee	Private		
	_getCurrentSupply	Private		
	_takeLiquidity	Private	✓	
	_takeBurn	Private	✓	
	claimTokens	Public	✓	onlyOwner
	calculateTaxFee	Private		
	calculateLiquidityFee	Private		
	removeAllFee	Private	✓	
	restoreAllFee	Private	✓	
	isExcludedFromFee	Public		-
	_approve	Private	✓	
	setTaxFee	Public	✓	onlyOwner
	setLiquidityFee	Public	✓	onlyOwner
	setBurnFee	Public	✓	onlyOwner
	setMarketingPerc	Public	✓	onlyOwner
	_transfer	Private	✓	
	swapAndMarketing	Private	✓	lockTheSwap
	swapAndLiquify	Private	✓	lockTheSwap
	swapTokensForEth	Private	✓	
	addLiquidity	Private	✓	
	_tokenTransfer	Private	✓	
	_transferStandard	Private	✓	
	_transferToExcluded	Private	✓	
	_transferFromExcluded	Private	✓	
	_transferBothExcluded	Private	✓	

Contract Flow



Domain Info

Domain Name	martiandoge.top
Registry Domain ID	D20220405G10001G_79160172-top
Creation Date	2022-04-05T14:20:30Z
Updated Date	2022-04-05T14:20:31Z
Registry Expiry Date	2023-04-05T14:20:30Z
Registrar WHOIS Server	whois.publicdomainregistry.com
Registrar URL	http://publicdomainregistry.com
Registrar	PDR Ltd
Registrar IANA ID	303

The domain has been created 1 day 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

Martian Doge Token is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like draining all the fees accumulated to the team wallet and blacklisting users from trading. The max amount of fees that can be set is 30%. A multi-wallet signing pattern will provide security against potential hacks. Temporarily locking the contract or renouncing ownership will eliminate all the contract threats.

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