Cyberscope

Audit Report

MafiaDoge

March 2022

Type	BEP20
Network	BSC
Address	0xFC1B2Eb26d1d18Ab01D405424C937Ae72364b1F4
Audited by	© cyberscope

Table of Contents

Contract Review

Contract Name	MAFIADOGE
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0xFC1B2Eb26d1d18Ab01D405424C937Ae72364b1F4
Symbol	MDoge
Decimals	9
Total Supply	1,000,000,000
Domain	mafiadoge.io

Source Files

Filename	**SHA256**
contract.sol	de09e94c670d003878471786de93958b81a5572078b4d32de80f5e25dd4d7d61

Audit Updates

Initial Audit	31st March 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#L380

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.

```
function manualswap() external {
    require(_msgSender() == _developmentAddress || _msgSender() ==
_marketingAddress || _msgSender() == owner());
    uint256 contractBalance = balanceOf(address(this));
    swapTokensForEth(contractBalance);
}

function manualsend() external {
    require(_msgSender() == _developmentAddress || _msgSender() ==
_marketingAddress || _msgSender() == owner());
    uint256 contractETHBalance = address(this).balance;
    sendETHToFee(contractETHBalance);
}
```

The contract sends funds to an address called liquidity receiver. Usually the contracts that add liquidity are using the swap and liquidy feature directly to the pair address.

```
_autoLiquidityReceiver.transfer(amount.div(2));
```

Recommendation

The contract could embody a check for the maximum amount of funds that can be swapped. Since a huge amount may violate 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
●	L05	Unused State Variable

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L119,125,195,199,203,207,215,220,224,229 and 5 more

Description

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

```
excludeMultipleAccountsFromFees
toggleSwap
setFee
setmarketingAddress
setdevAddress
transferFrom
approve
allowance
transfer
...
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality	minor
Location	contract.sol#L160,102

Description

Constant state variables should be declared constant to save gas.

```
_previousOwner
_autoLiquidityReceiver
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	contract.sol#L43,310,317,403,141,154,155,156

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.

```
_decimals
_symbol
_name
_tTotal
_feeSwap
marketingAddressUpdated
devAddressUpdated
WETH
```

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#L102,136

Description

There are segments that contain unused state variables.

```
_tOwned
_previousOwner
```

Recommendation

Remove unused state variables.

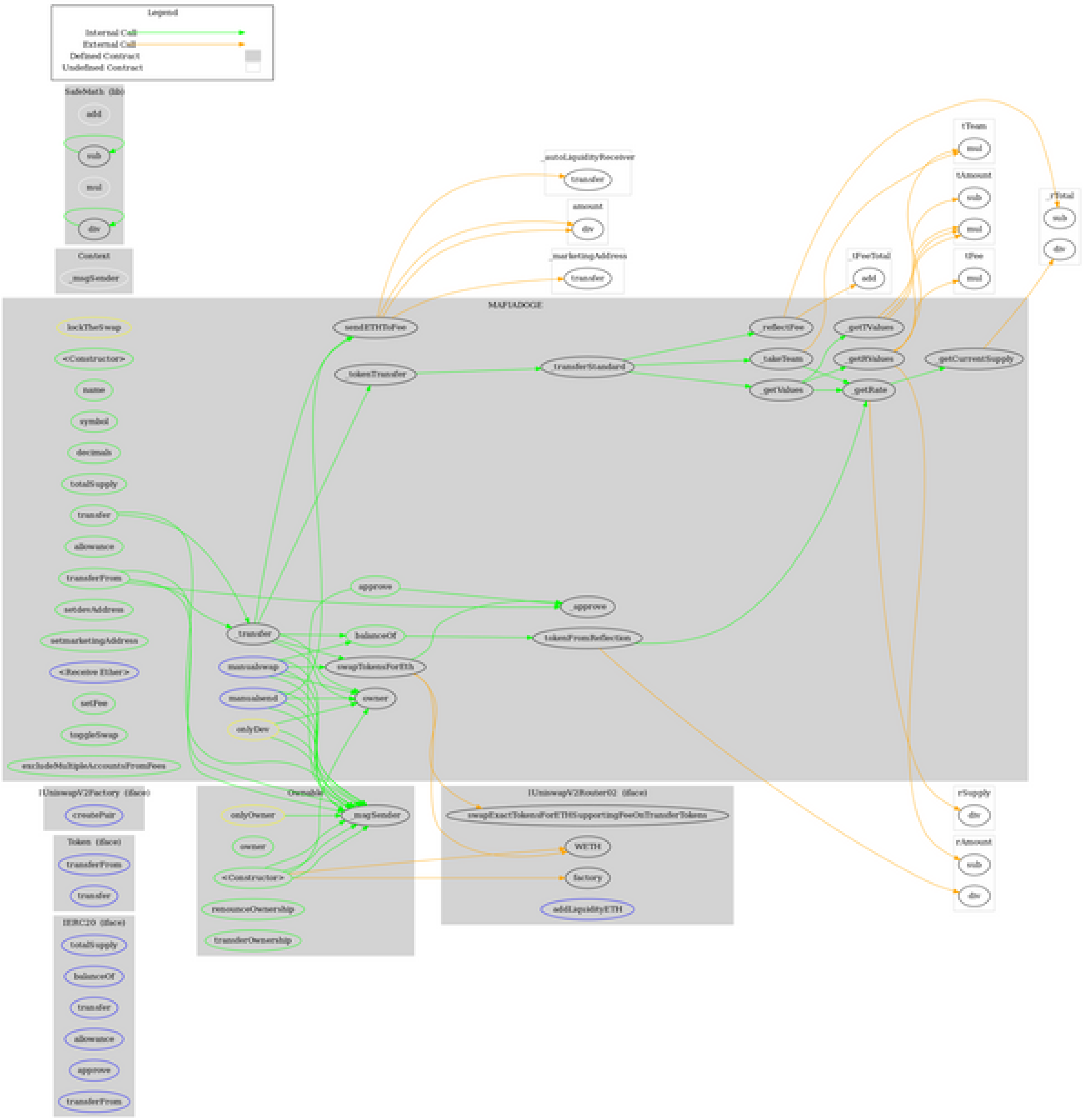
Contract Functions

Contract	**Type**	**Bases**		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Token	Interface			
	transferFrom	External	✓	-
	transfer	External	✓	-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
IUniswapV2Router02	Interface			
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
	factory	External		-
	WETH	External		-
	addLiquidityETH	External	Payable	-
Context	Implementation			
	_msgSender	Internal		
SafeMath	Library			
	add	Internal		
	sub	Internal		

	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
MAFIADOGE	Implementation	Context, IERC20, Ownable		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	tokenFromReflection	Private		
	_approve	Private	✓	
	_transfer	Private	✓	
	swapTokensForEth	Private	✓	lockTheSwap
	sendETHToFee	Private	✓	
	_tokenTransfer	Private	✓	
	setdevAddress	Public	✓	onlyDev
	setmarketingAddress	Public	✓	onlyDev
	_transferStandard	Private	✓	
	_takeTeam	Private	✓	
	_reflectFee	Private	✓	
	<Receive Ether>	External	Payable	-

	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	manualswap	External	✓	-
	manualsend	External	✓	-
	setFee	Public	✓	onlyDev
	toggleSwap	Public	✓	onlyDev
	excludeMultipleAccountsFromFees	Public	✓	onlyDev

Contract Flow



Domain Info

Domain Name	mafiadoge.io
Registry Domain ID	469c135f5e054e4c8644d8f83777b942-DONUTS
Creation Date	2022-03-29T10:25:31Z
Updated Date	2022-03-29T15:42:24Z
Registry Expiry Date	2023-03-29T10:25:31Z
Registrar WHOIS Server	whois.advancedregistrar.com
Registrar URL	http://www.netearthone.com
Registrar	NetEarth One Inc. dba NetEarth
Registrar IANA ID	1005

The domain has been created 2 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

MafiaDoge 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 sends funds to an address called “_autoLiquidityReceiver” that is misleading the LP auto-generation feature. This contract sends all the funds to the team wallets. 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 16% 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