



Cyberscope

Audit Report

MetaDogeBattle

March 2022

Type BEP20

Network BSC

Address 0x0F89644714c61A4072CAB7A6116780a1d70aEDf4

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	2
Audit Updates	2
Contract Analysis	3
ULTW - Unlimited Liquidity to Team Wallet	4
Description	4
Recommendation	4
Contract Diagnostics	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
L05 - Unused State Variable	9
Description	9
Recommendation	9
L04 - Conformance to Solidity Naming Conventions	10
Description	10
Recommendation	10
Contract Functions	11
Contract Flow	14
Domain Info	15
Summary	16
Disclaimer	17
About Cyberscope	18

Contract Review

Contract Name	MetaDogeBattle
Compiler Version	v0.8.4+commit.c7e474f2
Optimization	200 runs
Licence	None
Explorer	https://bscscan.com/token/0x0F89644714c61A4072CAB7A6116780a1d70aEDf4
Symbol	MDOB
Decimals	18
Total Supply	1,000,000,000
Source	contract.sol
Domain	

Audit Updates

Initial Audit	14th 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#L367

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 owner may take advantage of it by calling sequentially the *manualSwap* and *manualSend* functions.

```
if (!inSwap && from != uniswapV2Pair && swapEnabled && contractTokenBalance > 0)
{
    swapTokensForEth(contractTokenBalance);
    uint256 contractETHBalance = address(this).balance;
    if(contractETHBalance > 0) {
        sendETHToFee(address(this).balance);
    }
}
```

```
function manualSwap() external onlyOwner {
    uint256 contractBalance = balanceOf(address(this));
    swapTokensForEth(contractBalance);
}

function manualSend() external onlyOwner {
    uint256 contractETHBalance = address(this).balance;
    sendETHToFee(contractETHBalance);
}
```

Recommendation

The contract could embody a check for not allowing swapping many tokens once, since this may cause a price volatility.

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
●	L05	Unused State Variable
●	L04	Conformance to Solidity Naming Conventions

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L119,125,189,193,197,201,209,214,218,223 and 5 more

Description

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

```
excludeMultipleAccountsFromFees
toggleSwap
setFee
changeRouterVersion
setNewDevAddress
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#L102

Description

Constant state variables should be declared constant to save gas.

```
_previousOwner
```

Recommendation

Add the constant attribute to state variables that never change.

L05 - Unused State Variable

Criticality	minor
Location	contract.sol#L102

Description

There are segments that contain unused state variables.

```
_previousOwner
```

Recommendation

Remove unused state variables.

L04 - Conformance to Solidity Naming Conventions

Criticality

minor

Location

contract.sol#L43,296,384,140,144,146,150,151,152

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  
_taxFeeOnSell  
_taxFeeOnBuy  
_tTotal  
_swapEnabled  
devAddressUpdated  
WETH
```

Recommendation

Follow the Solidity naming convention.

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

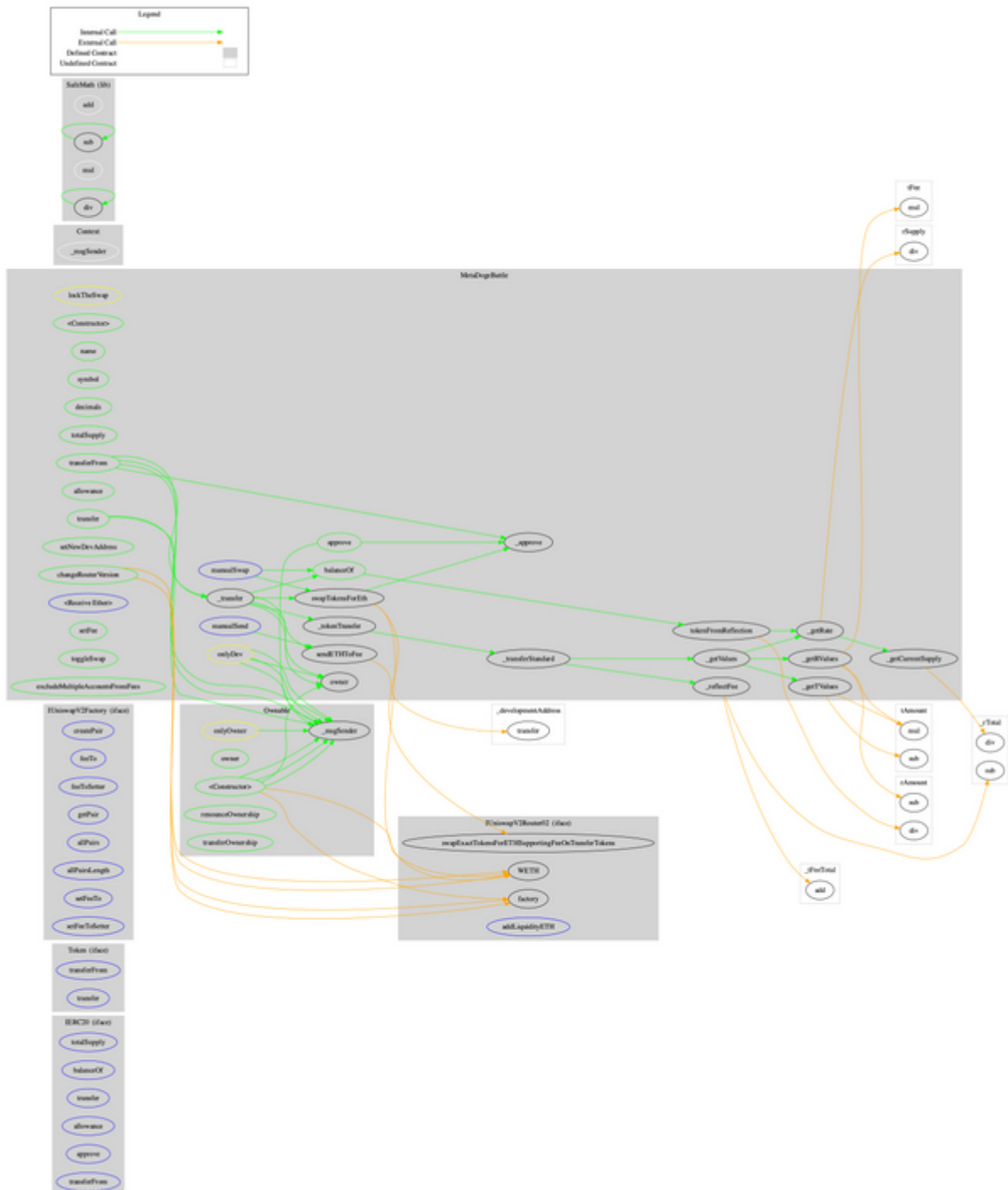
Contract Functions

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	✓	-
	feeTo	External		-
	feeToSetter	External		-
	getPair	External		-
	allPairs	External		-
	allPairsLength	External		-
	setFeeTo	External	✓	-
	setFeeToSetter	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
MetaDogeBattle	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	✓	
	setNewDevAddress	Public	✓	onlyDev
	_transferStandard	Private	✓	
	_reflectFee	Private	✓	
	changeRouterVersion	Public	✓	onlyOwner
	<Receive Ether>	External	Payable	-
	_getValues	Private		
	_getTValues	Private		
	_getRValues	Private		
	_getRate	Private		
	_getCurrentSupply	Private		
	manualSwap	External	✓	onlyOwner
	manualSend	External	✓	onlyOwner
	setFee	Public	✓	onlyDev
	toggleSwap	Public	✓	onlyDev
	excludeMultipleAccountsFromFees	Public	✓	onlyDev

Contract Flow



Domain Info

Domain Name	metadogebattle.io
Registry Domain ID	4e44f27758614169bbc1db042dd35994-DONUTS
Creation Date	2022-01-04T08:52:13Z
Updated Date	2022-01-14T07:03:59Z
Registry Expiry Date	2023-01-04T08:52:13Z
Registrar WHOIS Server	whois.pavietnam.vn
Registrar URL	http://www.pavietnam.vn
Registrar	P.A Viet Nam Company Limited
Registrar IANA ID	1649

The domain has been created 2 months before the creation of the audit. It will expire in 10 months.

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

Summary

The Smart Contract analysis reported no compiler error or critical issues. The contract owner has the ability to transfer a lot of funds to the team's wallet. Other than that, 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 11% fees both on sales and buys.

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>