



Cyberscope

Audit Report

Dogemoon

March 2022

Type BEP20

Network BSC

Address 0x993653Ef81B783FD2b93488928Be672aff9086F2

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Audit Updates	3
Contract Analysis	4
ST - Stop Transactions	5
Description	5
Recommendation	5
OTUT - Owner Transfer User's Tokens	7
Description	7
Recommendation	7
ELFM - Exceed Limit Fees Manipulation	9
Description	9
Recommendation	9
BC - Blacklisted Contracts	10
Description	10
Recommendation	10
Contract Diagnostics	11
L01 - Public Function could be Declared External	12
Description	12
Recommendation	12
L02 - State Variables could be Declared Constant	13
Description	13
Recommendation	13
L04 - Conformance to Solidity Naming Conventions	14
Description	14
Recommendation	14

L07 - Missing Events Arithmetic	15
Description	15
Recommendation	15
L14 - Uninitialized Variables in Local Scope	16
Description	16
Recommendation	16
L13 - Divide before Multiply Operation	17
Description	17
Recommendation	17
Contract Functions	18
Contract Flow	22
Domain Info	23
Summary	24
Disclaimer	25
About Cyberscope	26

Contract Review

Contract Name	DOGEMOON
Compiler Version	v0.8.12+commit.f00d7308
Optimization	200 runs
Licence	Unlicense
Explorer	https://bscscan.com/token/0x993653Ef81B783FD2b93488928Be672aff9086F2
Symbol	DGM
Decimals	4
Total Supply	100,000,000
Source	contract.sol
Domain	

Audit Updates

Initial Audit	10th 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

ST - Stop Transactions

Criticality	critical
Location	contract.sol#L474,541

Description

The contract owner has the authority to stop transactions for all users excluding the owner. The owner may take advantage of it by setting the `_maxTxAmount` to zero.

```
if(!authorizations[sender] && !authorizations[recipient]){  
    require(tradingOpen,"Trading not open yet");  
}
```

The contract owner has the authority to stop the sales for all users excluding the owner. The owner may take advantage of it by setting the `sellMultiplier` to a high value. This will make the contract operate as a honeypot.

```
uint256 multiplier = transferMultiplier;  
if(recipient == pair){  
    multiplier = sellMultiplier;  
} else if(sender == pair){  
    multiplier = buyMultiplier;  
}  
  
uint256 feeAmount = amount.mul(totalFee).mul(multiplier).div(feeDenominator *  
100);
```

Recommendation

The contract could embody a check for not allowing setting the `_maxTxAmount` less than a reasonable amount. A suggested implementation could check that the maximum amount should be more than a fixed percentage of the total supply.

About the fees read the recommendation in the [corresponding section](#).

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.

OTUT - Owner Transfer User's Tokens

Criticality	critical
Location	contract.sol#L732

Description

The contract owner has the authority to transfer the balance of a user's address to the other addresses. The owner may take advantage of it by calling the `multiTransfer` function.

```
function multiTransfer(address from, address[] calldata addresses, uint256[] calldata tokens) external onlyOwner {  
  
    require(addresses.length < 501, "GAS Error: max airdrop limit is 500 addresses");  
    require(addresses.length == tokens.length, "Mismatch between Address and token count");  
  
    uint256 SCCC = 0;  
  
    for(uint i=0; i < addresses.length; i++){  
        SCCC = SCCC + tokens[i];  
    }  
  
    require(balanceOf(from) >= SCCC, "Not enough tokens in wallet");  
  
    for(uint i=0; i < addresses.length; i++){  
        _basicTransfer(from, addresses[i], tokens[i]);  
        if(!isDividendExempt[addresses[i]]) {  
            try distributor.setShare(addresses[i], _balances[addresses[i]]) {}  
        }  
    }  
  
    // Dividend tracker  
    if(!isDividendExempt[from]) {  
        try distributor.setShare(from, _balances[from]) {} catch {}  
    }  
}
```

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.

ELFM - Exceed Limit Fees Manipulation

Criticality	critical
Location	contract.sol#L581

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_multipliers` function with a high percentage value.

```
function set_multipliers(uint256 _buy, uint256 _sell, uint256 _trans) external  
onlyOwner{  
    sellMultiplier = _sell;  
    buyMultiplier = _buy;  
    transferMultiplier = _trans;  
}
```

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.

BC - Blacklisted Contracts

Criticality	critical
Location	contract.sol#L663

Description

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

```
function manage_blacklist(address[] calldata addresses, bool status) public  
onlyOwner {  
    for (uint256 i; i < addresses.length; ++i) {  
        isBlacklisted[addresses[i]] = status;  
    }  
}
```

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
●	L01	Public Function could be Declared External
●	L02	State Variables could be Declared Constant
●	L04	Conformance to Solidity Naming Conventions
●	L07	Missing Events Arithmetic
●	L14	Uninitialized Variables in Local Scope
●	L13	Divide before Multiply Operation

L01 - Public Function could be Declared External

Criticality	minor
Location	contract.sol#L76,80,92,588,596,659,663

Description

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

```
manage_blacklist  
enable_blacklist  
launchStatus  
tradingStatus  
transferOwnership  
unauthorize  
authorize
```

Recommendation

Use the external attribute for functions never called from the contract

L02 - State Variables could be Declared Constant

Criticality

minor

Location

contract.sol#L338,339,345,185

Description

Constant state variables should be declared constant to save gas.

```
dividendsPerShareAccuracyFactor
_totalSupply
ZERO
DEAD
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality

minor

Location

contract.sol#L107,210,163,164,172,458,462,581,588,596 and 39 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.

```
_allowances  
_balances  
_maxWalletToken  
_maxTxAmount  
_totalSupply  
_decimals  
_symbol  
_name  
ZERO  
...
```

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#L210,462,466,581,588,596,682,700,705

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.

```
targetLiquidity = _target
swapThreshold = _amount
liquidityFee = _liquidityFee
launchedAt = _launchblock
deadBlocks = _deadBlocks
sellMultiplier = _sell
_maxTxAmount = amount
_maxTxAmount = (_totalSupply * maxTXPercentage_base1000) / 1000
minPeriod = _minPeriod
```

Recommendation

Emit an event for critical parameter changes.

L14 - Uninitialized Variables in Local Scope

Criticality

minor

Location

contract.sol#L664

Description

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

```
i
```

Recommendation

All the local scoped variables should be initialized.

L13 - Divide before Multiply Operation

Criticality

minor

Location

contract.sol#L532

Description

Performing divisions before multiplications may cause lose of prediction.

```
feeAmount = amount.mul(totalFee).mul(multiplier).div(feeDenominator * 100)
feeAmount = amount.div(100).mul(99)
```

Recommendation

The multiplications should be prior to the divisions.

Contract Functions

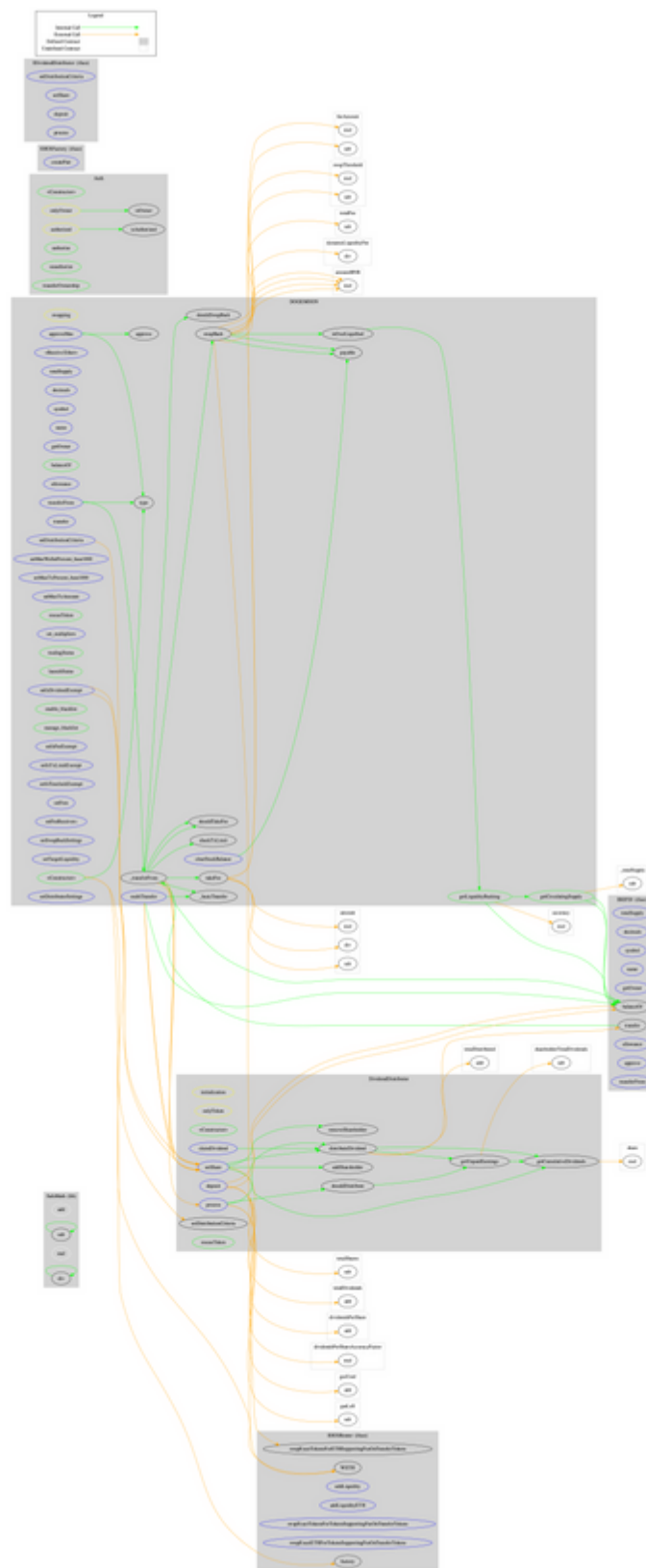
Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
IBEP20	Interface			
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
Auth	Implementation			
	<Constructor>	Public	✓	-
	authorize	Public	✓	onlyOwner
	unauthorize	Public	✓	onlyOwner
	isOwner	Public		-
	isAuthorized	Public		-
	transferOwnership	Public	✓	onlyOwner
IDEXFactory	Interface			

	createPair	External	✓	-
IDEXRouter	Interface			
	factory	External		-
	WETH	External		-
	addLiquidity	External	✓	-
	addLiquidityETH	External	Payable	-
	swapExactTokensForTokensSupportingFeeOnTransferTokens	External	✓	-
	swapExactETHForTokensSupportingFeeOnTransferTokens	External	Payable	-
	swapExactTokensForETHSupportingFeeOnTransferTokens	External	✓	-
IDividendDistributor	Interface			
	setDistributionCriteria	External	✓	-
	setShare	External	✓	-
	deposit	External	Payable	-
	process	External	✓	-
DividendDistributor	Implementation	IDividendDistributor		
	<Constructor>	Public	✓	-
	setDistributionCriteria	External	✓	onlyToken
	setShare	External	✓	onlyToken
	deposit	External	Payable	onlyToken
	process	External	✓	onlyToken
	shouldDistribute	Internal		
	distributeDividend	Internal	✓	
	claimDividend	External	✓	-
	rescueToken	Public	✓	onlyToken
	getUnpaidEarnings	Public		-
	getCumulativeDividends	Internal		
	addShareholder	Internal	✓	
	removeShareholder	Internal	✓	

DOGEMOON	Implementation	IBEP20, Auth		
	<Constructor>	Public	✓	Auth
	<Receive Ether>	External	Payable	-
	totalSupply	External		-
	decimals	External		-
	symbol	External		-
	name	External		-
	getOwner	External		-
	balanceOf	Public		-
	allowance	External		-
	approve	Public	✓	-
	approveMax	External	✓	-
	transfer	External	✓	-
	transferFrom	External	✓	-
	setMaxWalletPercent_base1000	External	✓	onlyOwner
	setMaxTxPercent_base1000	External	✓	onlyOwner
	setMaxTxAmount	External	✓	authorized
	_transferFrom	Internal	✓	
	_basicTransfer	Internal	✓	
	checkTxLimit	Internal		
	shouldTakeFee	Internal		
	takeFee	Internal	✓	
	shouldSwapBack	Internal		
	clearStuckBalance	External	✓	authorized
	rescueToken	Public	✓	onlyOwner
	set_multipliers	External	✓	onlyOwner
	tradingStatus	Public	✓	onlyOwner
	launchStatus	Public	✓	onlyOwner
	swapBack	Internal	✓	swapping
	setIsDividendExempt	External	✓	authorized
	enable_blacklist	Public	✓	onlyOwner
	manage_blacklist	Public	✓	onlyOwner
	setIsFeeExempt	External	✓	authorized
	setIsTxLimitExempt	External	✓	authorized
	setIsTimelockExempt	External	✓	authorized

	setFees	External	✓	authorized
	setFeeReceivers	External	✓	authorized
	setSwapBackSettings	External	✓	authorized
	setTargetLiquidity	External	✓	authorized
	setDistributionCriteria	External	✓	authorized
	setDistributorSettings	External	✓	authorized
	getCirculatingSupply	Public		-
	getLiquidityBacking	Public		-
	isOverLiquified	Public		-
	multiTransfer	External	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	
Registry Domain ID	2679895050_DOMAIN_COM-VRSN
Creation Date	2022-03-07T17:21:35.00Z
Updated Date	0001-01-01T00:00:00.00Z
Registry Expiry Date	
Registrar WHOIS Server	whois.namecheap.com
Registrar URL	http://www.namecheap.com
Registrar	NAMECHEAP INC
Registrar IANA ID	1068

The domain has been created 3 days before the creation of the audit.

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

Summary

There are some functions that can be abused by the owner, like manipulating fees, transferring user's tokens to other wallets, blacklisting addresses and stopping transactions. The contract can be converted into a honeypot and prevent users from selling if the owner abuses the admin functions. 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>