



Cyberscope

Audit Report

UDOGE

April 2022

Type BEP20

Network BSC

Address 0xaa8681735892983eb882af5c2821a3147e9fc27e

Audited by © cyberscope

Table of Contents

Table of Contents	1
Contract Review	3
Source Files	3
Audit Updates	4
Contract Analysis	5
ELFM - Exceed Limit Fees Manipulation	6
Description	6
Recommendation	6
BC - Blacklisted Contracts	7
Description	7
Recommendation	7
Contract Diagnostics	8
L01 - Public Function could be Declared External	9
Description	9
Recommendation	9
L02 - State Variables could be Declared Constant	10
Description	10
Recommendation	10
L04 - Conformance to Solidity Naming Conventions	11
Description	11
Recommendation	11
L07 - Missing Events Arithmetic	12
Description	12
Recommendation	12
L12 - Using Variables before Declaration	13
Description	13

Recommendation	13
L14 - Uninitialized Variables in Local Scope	14
Description	14
Recommendation	14
Contract Functions	15
Contract Flow	22
Domain Info	23
Summary	24
Disclaimer	25
About Cyberscope	26

Contract Review

Contract Name	UDogeToken
Compiler Version	v0.6.12+commit.27d51765
Optimization	200 runs
Licence	MIT
Explorer	https://bscscan.com/token/0xAA8681735892983Eb882AF5C2821A3147E9fC27E
Symbol	UDoge
Decimals	18
Total Supply	1,000,000,000
Domain	udoge.org

Source Files

Filename	SHA256
Context.sol	2fc3d839278ddcc7ecbc9d7cc465530b02f9e8afdc6e345ed99cc55f34f176d5
DividendPayingToken.sol	0644a56320c9c6c6913ffb94f9754a116c24c5c0d2d3a4b621333451a5089a46
DividendPayingTokenInterface.sol	f27b9720bb802c6c41f827d4b32a8a5506b1aa432e6dabd405af300df3972111
DividendPayingTokenOptionalInterface.sol	5218929fb8f48de47326037c158b97b5a5fa4147b30690c28cb1146593b4fafc
ERC20.sol	4e4abd2ad56c6b7b9fc0d8cba74d312db0b0efc97b931e807024f13ec7e6eca3

IERC20.sol	8f6d61c50e08e330da0ae3135c3797fe9b5f20f9925550f8855c52437b38db59
IERC20Metadata.sol	07b046873251c771868686065501ac0c082926ec1306d4ad254094c6cd99a36b
IterableMapping.sol	75c825a4c01934903b4eed6787f5bc08a71f941011c405118d6d0b8d030486b5
IUniswapV2Factory.sol	0fe06a8272d951beb6db9cc61dcbf155d64b274134b226e48f67c478fd8cac0b
IUniswapV2Pair.sol	55b67598c763cd150b9cdc87c0502f12f0ef411ae349d635947cf22fb1bb2a5b
IUniswapV2Router.sol	5305b4e50768a4f5631b443d1ea3fb2bde3c451e375ea098aeb5f97145b71d0
Ownable.sol	24b9b093bcb63ea7fdd5eb4e7a65c787f35bf4ad6f510862e1fe8b78e2d27a80
SafeMath.sol	5c2d075018ba0852d43e8da477965f3edd78ea0e78e148fddcaa6eb6a4b1e919
SafeMathInt.sol	9bb2b51fd4fae5e705a25e5712163a516b61b0db5accefcb5c1959a16978040
SafeMathUint.sol	005c4b99212ed30491ca1028f860090aa693237323eb98a422c3e1646db97c6d
UDogeToken.sol	0490b127be2fc79c78cf355ddf30c61d85f31774c4b89f17ba54ef009c62c0ee

Audit Updates

Initial Audit	3rd 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	critical
Location	contract.sol#L175,180

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 `setMarketingFee` function with a high percentage value.

```
function setMarketingFee(uint256 value) external onlyOwner{
    marketingFee = value;
    totalFees = USDTRewardsFee.add(liquidityFee).add(marketingFee);
}
```

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	medium
Location	contract.sol#L193

Description

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

```
require(!_isBlacklisted[from] && !_isBlacklisted[to], 'Blacklisted address');
```

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
●	L12	Using Variables before Declaration
●	L14	Uninitialized Variables in Local Scope

L01 - Public Function could be Declared External

Criticality

minor

Location

UDogeToken.sol#L121,138,154,187,210,229,233,237,575,619

Description

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

```
process
getAccountAtIndex
dividendTokenBalanceOf
withdrawableDividendOf
isExcludedFromFees
updateGasForProcessing
setAutomatedMarketMakerPair
excludeMultipleAccountsFromFees
updateUniswapV2Router
...
```

Recommendation

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

L02 - State Variables could be Declared Constant

Criticality

minor

Location

UDogeToken.sol#L23

Description

Constant state variables should be declared constant to save gas.

```
deadWallet
```

Recommendation

Add the constant attribute to state variables that never change.

L04 - Conformance to Solidity Naming Conventions

Criticality	minor
Location	UDogeToken.sol#L25,29,31,36,530

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.

```
_account  
_marketingWalletAddress  
USDTRewardsFee  
_isBlacklisted  
USDT
```

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	UDogeToken.sol#L166,171,175,180

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.

```
marketingFee = value  
liquidityFee = value  
swapTokensAtAmount = value  
USDTRewardsFee = value
```

Recommendation

Emit an event for critical parameter changes.

L12 - Using Variables before Declaration

Criticality

minor

Location

UDogeToken.sol#L358

Description

The contract is using a variable before the declaration. This is usually happening either if it has not been declared yet or the variable has been declared in a different scope.

```
lastProcessedIndex  
iterations  
claims
```

Recommendation

The variables should be declared before any usage of them.

L14 - Uninitialized Variables in Local Scope

Criticality

minor

Location

UDogeToken.sol#L358

Description

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

`lastProcessedIndex``claims``iterations`

Recommendation

All the local scoped variables should be initialized.

Contract Functions

Contract	Type	Bases		
	Function Name	Visibility	Mutability	Modifiers
Context	Implementation			
	_msgSender	Internal		
	_msgData	Internal		
DividendPayingToken	Implementation	ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface		
	<Constructor>	Public	✓	ERC20
	distributeCAKEDividends	Public	✓	onlyOwner
	withdrawDividend	Public	✓	-
	_withdrawDividendOfUser	Internal	✓	
	dividendOf	Public		-
	withdrawableDividendOf	Public		-
	withdrawnDividendOf	Public		-
	accumulativeDividendOf	Public		-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_setBalance	Internal	✓	
DividendPayingTokenInterface	Interface			
	dividendOf	External		-
	withdrawDividend	External	✓	-

DividendPayingTokenOptionallInterface	Interface			
	withdrawableDividendOf	External		-
	withdrawnDividendOf	External		-
	accumulativeDividendOf	External		-
ERC20	Implementation	Context, IERC20, IERC20Metadata		
	<Constructor>	Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_mint	Internal	✓	
	_burn	Internal	✓	
	_approve	Internal	✓	
	_beforeTokenTransfer	Internal	✓	
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-

IERC20Metadata	Interface	IERC20		
	name	External		-
	symbol	External		-
	decimals	External		-
IterableMapping	Library			
	get	Public		-
	getIndexOfKey	Public		-
	getKeyAtIndex	Public		-
	size	Public		-
	set	Public	✓	-
	remove	Public	✓	-
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	✓	-

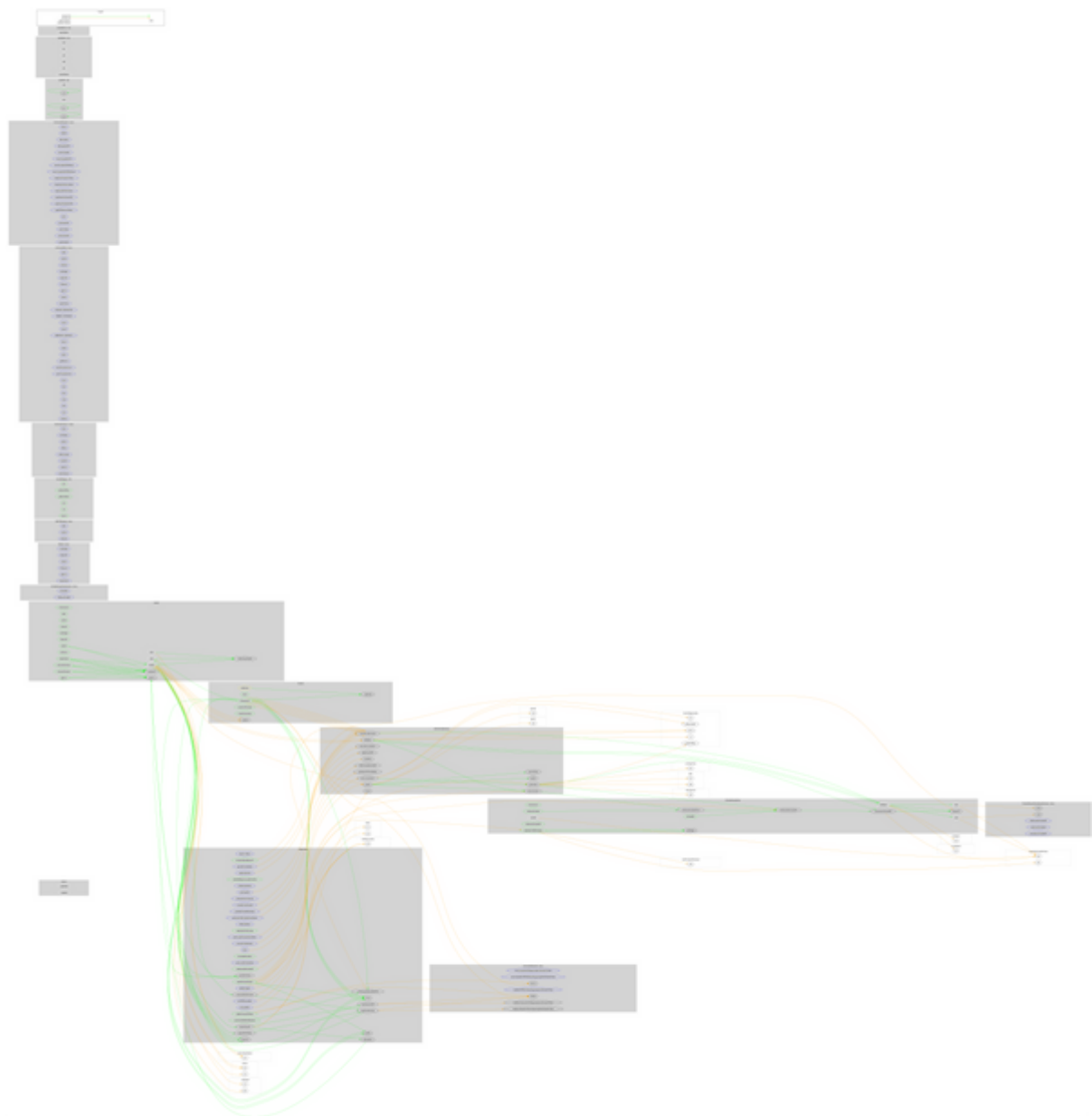
	DOMAIN_SEPARATOR	External		-
	PERMIT_TYPEHASH	External		-
	nonces	External		-
	permit	External	✓	-
	MINIMUM_LIQUIDITY	External		-
	factory	External		-
	token0	External		-
	token1	External		-
	getReserves	External		-
	price0CumulativeLast	External		-
	price1CumulativeLast	External		-
	kLast	External		-
	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	✓	-
Ownable	Implementation	Context		
	<Constructor>	Public	✓	-
	owner	Public		-
	renounceOwnership	Public	✓	onlyOwner
	transferOwnership	Public	✓	onlyOwner
SafeMath	Library			
	add	Internal		
	sub	Internal		
	sub	Internal		
	mul	Internal		
	div	Internal		
	div	Internal		
	mod	Internal		
	mod	Internal		
SafeMathInt	Library			
	mul	Internal		
	div	Internal		

	sub	Internal		
	add	Internal		
	abs	Internal		
	toUint256Safe	Internal		
SafeMathUint	Library			
	toInt256Safe	Internal		
UDogeToken	Implementation	ERC20, Ownable		
	<Constructor>	Public	✓	ERC20
	<Receive Ether>	External	Payable	-
	updateDividendTracker	Public	✓	onlyOwner
	updateUniswapV2Router	Public	✓	onlyOwner
	excludeFromFees	Public	✓	onlyOwner
	excludeMultipleAccountsFromFees	Public	✓	onlyOwner
	setMarketingWallet	External	✓	onlyOwner
	setUSDTRewardsFee	External	✓	onlyOwner
	setSwapTokensAtAmount	External	✓	onlyOwner
	setLiquiditFee	External	✓	onlyOwner
	setMarketingFee	External	✓	onlyOwner
	setAutomatedMarketMakerPair	Public	✓	onlyOwner
	blacklistAddress	External	✓	onlyOwner
	_setAutomatedMarketMakerPair	Private	✓	
	updateGasForProcessing	Public	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getClaimWait	External		-
	getTotalDividendsDistributed	External		-
	isExcludedFromFees	Public		-
	withdrawableDividendOf	Public		-
	dividendTokenBalanceOf	Public		-
	excludeFromDividends	External	✓	onlyOwner
	getAccountDividendsInfo	External		-
	getAccountDividendsInfoAtIndex	External		-
	processDividendTracker	External	✓	-
	claim	External	✓	-

	getLastProcessedIndex	External		-
	getNumberOfDividendTokenHolders	External		-
	_transfer	Internal	✓	
	swapAndSendToFee	Private	✓	
	swapAndLiquify	Private	✓	
	swapTokensForEth	Private	✓	
	swapTokensForCake	Private	✓	
	addLiquidity	Private	✓	
	swapAndSendDividends	Private	✓	
USDTDividend Tracker	Implementation	Ownable, DividendPayingToken		
	<Constructor>	Public	✓	DividendPayingToken
	_transfer	Internal	✓	
	withdrawDividend	Public	✓	-
	excludeFromDividends	External	✓	onlyOwner
	updateClaimWait	External	✓	onlyOwner
	getLastProcessedIndex	External		-
	getNumberOfTokenHolders	External		-
	getAccount	Public		-
	getAccountAtIndex	Public		-
	canAutoClaim	Private		
	setBalance	External	✓	onlyOwner
	process	Public	✓	-
	processAccount	Public	✓	onlyOwner

Contract Flow



Domain Info

Domain Name	udoge.org
Registry Domain ID	D402200000019438639-LROR
Creation Date	2022-03-28T07:36:08Z
Updated Date	2022-03-30T10:15:41Z
Registry Expiry Date	2023-03-28T07:36:08Z
Registrar WHOIS Server	whois.registrar.amazon.com
Registrar URL	registrar.amazon.com
Registrar	Amazon Registrar, Inc.
Registrar IANA ID	468

The domain has been created 6 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

UDoge is an interesting project that has a friendly and growing community. There are some functions that can be abused by the owner, like manipulating fees up to 100% and blacklisting users from trading. 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>