Cyberscope

Audit Report

Doge Clone

June 2023

Network BSC

Address 0x5666D8Bc531f3214167661008a4066ef03C8F026

Audited by © cyberscope

Analysis

Critical Medium Minor / Informative Pass

Severity	Code	Description	Status
●	ST	Stops Transactions	Passed
●	OTUT	Transfers User's Tokens	Passed
●	ELFM	Exceeds Fees Limit	Passed
●	MT	Mints Tokens	Passed
●	BT	Burns Tokens	Passed
●	BC	Blacklists Addresses	Passed

Table of Contents

Review

Contract Name	DogeClone
Compiler Version	v0.8.18+commit.87f61d96
Optimization	200 runs
Explorer	https://bscscan.com/address/0x5666d8bc531f3214167661008a4066ef03c8f026
Address	0x5666d8bc531f3214167661008a4066ef03c8f026
Network	BSC
Symbol	DOGAMG
Decimals	18
Total Supply	15,000,000,000

Audit Updates

Initial Audit	07 Jun 2023

Source Files

Filename	SHA256
contracts/DogeClone.sol	3281d77cc6f5dea4f104db3685d3d5dff43d4ba11b2ede525286be3e9cb36115

MultiSig

The contract incorporates a custom multi-signature (multi-sig) functionality, utilizing a total of 7 immutable multi-sig wallets. These wallets cannot be altered. When executing an admin function, the contract maintains specific variables to track the proposed value and an array of multi-sig wallets that have endorsed the proposed modification. In order for the modification to be implemented, all 7 wallets must call the admin function with an identical value. If any of the wallets provide a different value, the array of approved wallets is reset, and the proposed value is updated to the new value. Once all 7 wallets unanimously approve the update with the same value, the contract proceeds with the modification.

Let'a assume the authorized wallets are the following:

- 0x956a99ACB42475C029F365FBfDc2765D3A953e9b
- 0xD782DEaC7F75C0129B0A88025D61b925c03a1f6e
- 0x740a6D2b6522031afE42C650aA1c1186492A96D0
- 0xBaeAfDade87f66710C23711e7Dddc2AB17dFD56E
- 0xa54C7bd8e067130F1e21e358B4badd25c68D7893
- 0x7eEbC93a6aef63101a7cfAceb5D0ec89Cc4D42BE
- 0x68cFe7566EEdb3Cc88055ed66882175E7B82F23e

A simple breakdown of the process when an authorized wallet calls an admin function like `setTaxFee` is described below:

1. The `setTaxFee` function is invoked by wallet `0x956a99ACB42475C029F365FBfDc2765D3A953e9b` with a value of 10 (0.1%).
2. The contract checks if the new tax fee for the tax fee modification is the same as the previous proposed tax fee (`proposedTaxFee`). If it's different, it means that either the `0x956a99ACB42475C029F365FBfDc2765D3A953e9b` is the first to propose a new tax fee or `0x956a99ACB42475C029F365FBfDc2765D3A953e9b` has suggested a new value.
3. If the new tax fee is different, the contract resets the array of approved wallets (`approvedTaxFeeSignatures`) and updates the proposed tax fee (`proposedTaxFee`) to the new value (`proposedTaxFee = 10`).
4. The `0x956a99ACB42475C029F365FBfDc2765D3A953e9b` that called the setTaxFee function is added to the array of approved wallets (`approvedTaxFeeSignatures`

). So `approvedTaxFeeSignatures = [0x956a99ACB42475C029F365FBfDc2765D3A953e9b]` .

5. Then wallet `0xD782DEaC7F75C0129B0A88025D61b925c03a1f6e` calls the setFee function with the same value, so it is added to the array of approved wallets and `approvedTaxFeeSignatures = [0x956a99ACB42475C029F365FBfDc2765D3A953e9b, 0xD782DEaC7F75C0129B0A88025D61b925c03a1f6e]` .
6. The process repeats until all 7 wallets approve the same value.
7. If all wallets have agreed upon the same value, the contract proceeds with the tax fee update.
8. If any wallet has provided a different value, the contract resets the array of approved wallets, updates the proposed tax fee (`proposedTaxFee`) to the new value, and the process starts again.

If an authorized wallet wants to reject the proposed tax fee, all it needs to do is call the `setTaxFee` function with a different value than the one proposed by another wallet.

The same approach is followed in the methods:

- setTaxAddress(address newTaxAddress)
- setTaxFee(uint8 newTaxFee)
- setTradeCooldown(uint8 newTradeCooldown)
- setIsExcludedFromFees(address account, bool excluded)
- setIsExcludedFromAntibot(address account, bool excluded)
- renounceOwnership()

The contract depicts which wallet addresses have approve the methods in the following arrays:

- getApprovedTaxAddressSignatures, `proposedTaxAddress`
- getApprovedTaxFeeSignatures, `proposedTaxFee`
- getApprovedTradeCooldownSignatures, `proposedTradeCooldown`
- getApprovedExcludedAddressSignatures, `proposedExcludedAddress, proposedExcludedAddressValue`
- getApprovedExcludedFromAntibotAddressSignatures, `proposedExcludedFromAntibotAddress, proposedExcludedFromAntibotAddressValue`

- getApprovedRenounceOwnershipSignatures

Roles

User

The user can interact with the following functions:

- `function name()`
- `function symbol()`
- `function decimals()`
- `function totalSupply()`
- `function balanceOf(address account)`
- `function transfer(address recipient, uint256 amount)`
- `function allowance(address account, address spender)`
- `function approve(address spender, uint256 amount)`
- `function transferFrom(address sender, address recipient, uint256 amount)`
- `function increaseAllowance(address spender, uint256 addedValue)`
- `function decreaseAllowance(address spender, uint256 subtractedValue)`
- `function getApprovedTaxAddressSignatures()`
- `function getApprovedTaxFeeSignatures()`
- `function getApprovedTradeCooldownSignatures()`
- `function getApprovedExcludedFromFeesAddressSignatures()`
- `function getApprovedExcludedFromAntibotAddressSignatures()`
- `function getApprovedRenounceOwnershipSignatures()`
- `function getAuthorizedWallets()`

Authorized

The authorized multi-sig wallets have authority over the following functions:

- `function setTaxAddress(address newTaxAddress)`
- `function setTaxFee(uint8 newTaxFee)`
- `function setTradeCooldown(uint8 newTradeCooldown)`
- `function setIsExcludedFromFees(address account, bool excluded)`

- `function setIsExcludedFromAntibot(address account, bool excluded)`
- `function renounceOwnership()`

Findings Breakdown

Critical	0
Medium	0
Minor / Informative	0

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	0	0	0	0
Medium	0	0	0	0
Minor / Informative	0	0	0	0

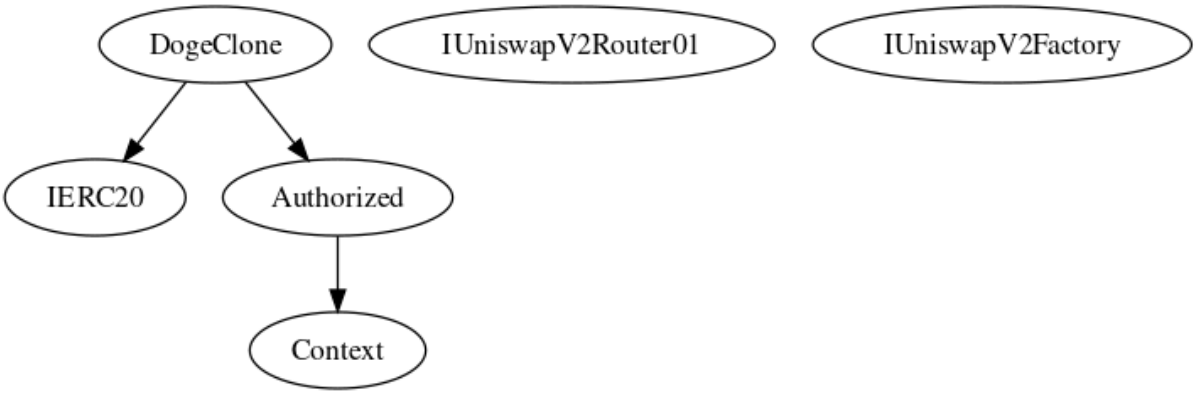
Functions Analysis

Contract	Type	Bases		
	Function Name	**Visibility**	**Mutability**	**Modifiers**
IERC20	Interface			
	totalSupply	External		-
	balanceOf	External		-
	transfer	External	✓	-
	allowance	External		-
	approve	External	✓	-
	transferFrom	External	✓	-
IUniswapV2Router01	Interface			
	factory	External		-
	WETH	External		-
IUniswapV2Factory	Interface			
	createPair	External	✓	-
Context	Implementation			
	_msgSender	Internal		

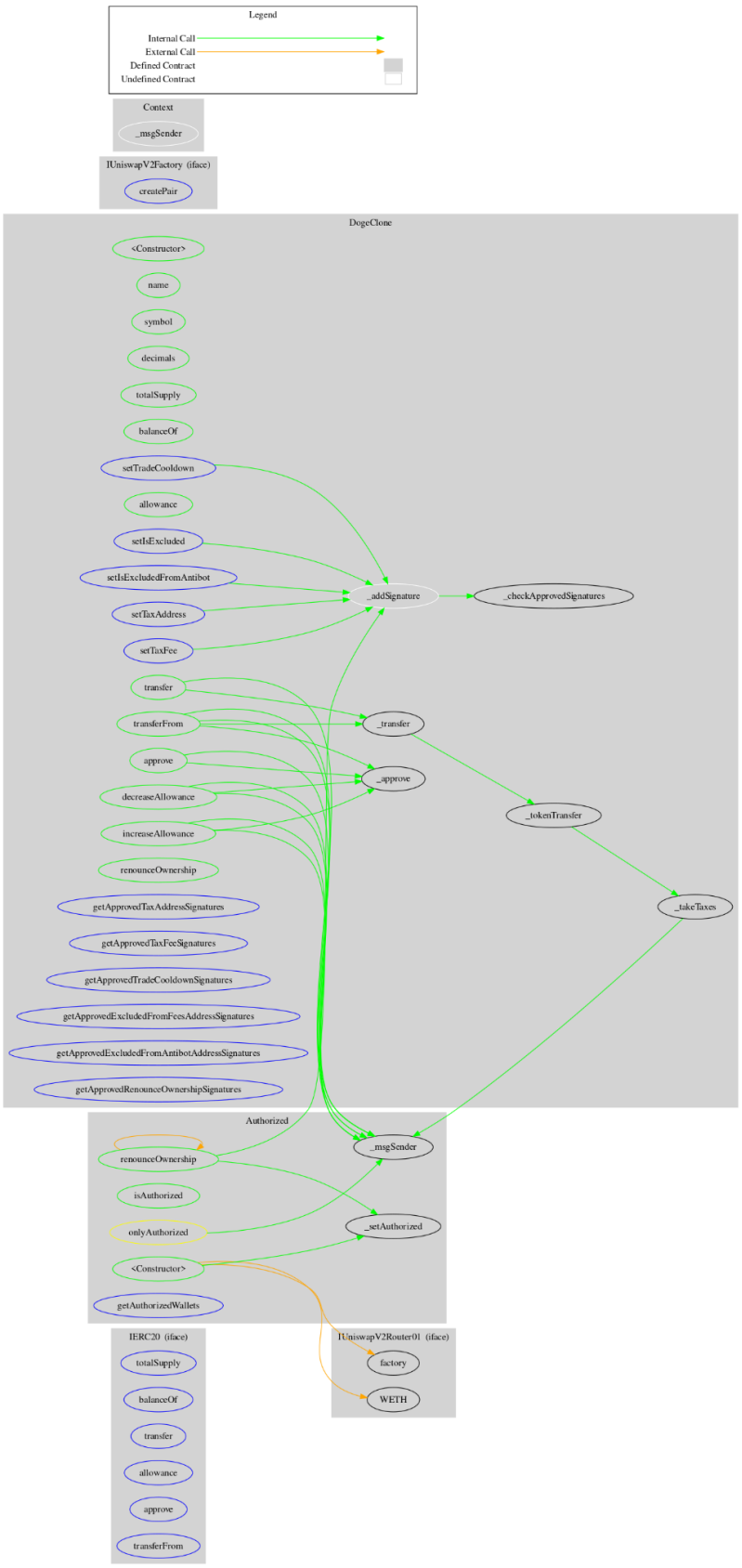
Authorized	Implementation	Context		
		Public	✓	-
	isAuthorized	Public		-
	renounceOwnership	Public	✓	onlyAuthorized
	_setAuthorized	Private	✓	
	getAuthorizedWallets	External		-
DogeClone	Implementation	IERC20, Authorized		
		Public	✓	-
	name	Public		-
	symbol	Public		-
	decimals	Public		-
	totalSupply	Public		-
	balanceOf	Public		-
	transfer	Public	✓	-
	allowance	Public		-
	approve	Public	✓	-
	transferFrom	Public	✓	-
	increaseAllowance	Public	✓	-
	decreaseAllowance	Public	✓	-
	_transfer	Internal	✓	
	_approve	Internal	✓	
	_tokenTransfer	Internal	✓	

	_takeTaxes	Internal	✓	
	_checkApprovedSignatures	Internal		
	_addSignature	Internal	✓	
	setTaxAddress	External	✓	onlyAuthorized
	setTaxFee	External	✓	onlyAuthorized
	setTradeCooldown	External	✓	onlyAuthorized
	setIsExcluded	External	✓	onlyAuthorized
	setIsExcludedFromAntibot	External	✓	onlyAuthorized
	renounceOwnership	Public	✓	onlyAuthorized
	getApprovedTaxAddressSignatures	External		-
	getApprovedTaxFeeSignatures	External		-
	getApprovedTradeCooldownSignatures	External		-
	getApprovedExcludedFromFeesAddressSignatures	External		-
	getApprovedExcludedFromAntibotAddressSignatures	External		-
	getApprovedRenounceOwnershipSignatures	External		-

Inheritance Graph



Flow Graph


Legend
Internal Call
External Call
Defined Contract
Undefined Contract
Context
_msgSender
IUniswapV2Factory (iface)
createPair
DogeClone
<Constructor>
name
symbol
decimals
totalSupply
balanceOf
setTradeCooldown
allowance
setIsExcluded
setIsExcludedFromAntibot
setTaxAddress
setTaxFee
transfer
transferFrom
approve
decreaseAllowance
increaseAllowance
renounceOwnership
getApprovedTaxAddressSignatures
getApprovedTaxFeeSignatures
getApprovedTradeCooldownSignatures
getApprovedExcludedFromFeesAddressSignatures
getApprovedExcludedFromAntibotAddressSignatures
getApprovedRenounceOwnershipSignatures
_addSignature
_checkApprovedSignatures
_transfer
_approve
_tokenTransfer
_takeTaxes
Authorized
renounceOwnership
isAuthorized
onlyAuthorized
<Constructor>
getAuthorizedWallets
_msgSender
_setAuthorized
IERC20 (iface)
totalSupply
balanceOf
transfer
allowance
approve
transferFrom
IUniswapV2Router01 (iface)
factory
WETH

Summary

Doge Clone contract implements a token mechanism. This audit investigates security issues, business logic concerns, and potential improvements. Doge Clone is an interesting project that has a friendly and growing community. The Smart Contract analysis reported no compiler errors or critical issues. The contract implements a multi-sig functionality. The contract multi-sig wallets 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 0.5% fees.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io