Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

August, 2021

Audit Details

Audited project

DogemonGo

Deployer address

0x637dEaD068d4874c88d957aE68EEd7C76A9c3f52

Client contacts:

DogemonGo team

Blockchain

Binance Smart Chain

Project website:

https://dogemongo.com

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (TechRate) owe no duty of care towards you or any other person, nor does TechRate make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and TechRate hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, TechRate hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against TechRate, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Background

TechRate was commissioned by DogemonGo to perform an audit of smart contracts:

https://bscscan.com/address/0x9e6b3e35c8f563b45d864f9ff697a144ad28a371#code

The purpose of the audit was to achieve the following:

- Ensure that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

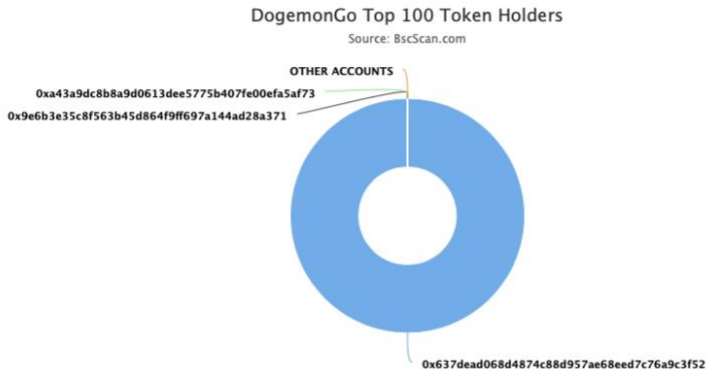
Token contract details for 02.08.2021

Contract name	DOGOToken
Contract address	0x9E6B3E35c8f563B45d864f9Ff697A144ad28A371
Total supply	100,000,000,000
Token ticker	DOGO
Decimals	18
Token holders	3
Transactions count	8
Top 100 holders dominance	100.00%
Liquidity fee	3
DOGE rewards fee	7
Marketing fee	5
Total fees	15
DOGE address	0xba2ae424d960c26247dd6c32edc70b295c744c43
Uniswap V2 pair	0xa7040bdcabecee39f571c8f2c8d0d73cc10d62db
Contract deployer address	0x637dEaD068d4874c88d957aE68EEd7C76A9c3f52
Contract's current owner address	0x637dEaD068d4874c88d957aE68EEd7C76A9c3f52

DogemonGo Token Distribution

The top 100 holders collectively own 100.00% (100,000,000,000.00 Tokens) of DogemonGo

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 3



(A total of 100,000,000,000.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000.00 token)

DogemonGo Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x637dead068d4874c88d957ae68eed7c76a9c3f52	99,999,999,900	100.0000%
2	0x9e6b3e35c8f563b45d864f9ff697a144ad28a371	50	0.0000%
3	0xa43a9dc8b8a9d0613dee5775b407fe00efa5af73	50	0.0000%

Contract functions details

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ [Int] IERC20Metadata (IERC20)
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals

+ Context
  - [Int] _msgSender
  - [Int] _msgData

+ [Lib] SafeMath
  - [Int] add
  - [Int] sub
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] div
  - [Int] mod
  - [Int] mod

+ ERC20 (Context, IERC20, IERC20Metadata)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Int] _transfer #
  - [Int] _mint #
  - [Int] _burn #
  - [Int] _approve #
  - [Int] _beforeTokenTransfer #

+ [Lib] SafeMathUint
  - [Int] toInt256Safe

+ [Lib] SafeMathInt
  - [Int] mul
  - [Int] div

  - [Int] sub
  - [Int] add
  - [Int] abs
  - [Int] toUint256Safe

+ [Int] DividendPayingTokenInterface
  - [Ext] dividendOf
  - [Ext] withdrawDividend #

+ [Int] DividendPayingTokenOptionalInterface
  - [Ext] withdrawableDividendOf
  - [Ext] withdrawnDividendOf
  - [Ext] accumulativeDividendOf

+ Ownable (Context)
  - [Pub] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner

+ DividendPayingToken (ERC20, Ownable, DividendPayingTokenInterface, DividendPayingTokenOptionalInterface)
  - [Pub] <Constructor> #
    - modifiers: ERC20
  - [Pub] distributeDOGEDividends #
    - modifiers: onlyOwner
  - [Pub] withdrawDividend #
  - [Int] _withdrawDividendOfUser #
  - [Pub] dividendOf
  - [Pub] withdrawableDividendOf
  - [Pub] withdrawnDividendOf
  - [Pub] accumulativeDividendOf
  - [Int] _transfer #
  - [Int] _mint #
  - [Int] _burn #
  - [Int] _setBalance #

+ [Lib] IterableMapping
  - [Pub] get
  - [Pub] getIndexOfKey
  - [Pub] getKeyAtIndex
  - [Pub] size
  - [Pub] set #
  - [Pub] remove #

+ [Int] IUniswapV2Pair
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] allowance
  - [Ext] approve #

  - [Ext] transfer #
  - [Ext] transferFrom #
  - [Ext] DOMAIN_SEPARATOR
  - [Ext] PERMIT_TYPEHASH
  - [Ext] nonces
  - [Ext] permit #
  - [Ext] MINIMUM_LIQUIDITY
  - [Ext] factory
  - [Ext] token0
  - [Ext] token1
  - [Ext] getReserves
  - [Ext] price0CumulativeLast
  - [Ext] price1CumulativeLast
  - [Ext] kLast
  - [Ext] mint #
  - [Ext] burn #
  - [Ext] swap #
  - [Ext] skim #
  - [Ext] sync #
  - [Ext] initialize #

+ [Int] IUniswapV2Factory
  - [Ext] feeTo
  - [Ext] feeToSetter
  - [Ext] migrator
  - [Ext] getPair
  - [Ext] allPairs
  - [Ext] allPairsLength
  - [Ext] createPair #
  - [Ext] setFeeTo #
  - [Ext] setFeeToSetter #
  - [Ext] setMigrator #

+ [Int] IUniswapV2Router01
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidity #
  - [Ext] addLiquidityETH ($)
  - [Ext] removeLiquidity #
  - [Ext] removeLiquidityETH #
  - [Ext] removeLiquidityWithPermit #
  - [Ext] removeLiquidityETHWithPermit #
  - [Ext] swapExactTokensForTokens #
  - [Ext] swapTokensForExactTokens #
  - [Ext] swapExactETHForTokens ($)
  - [Ext] swapTokensForExactETH #
  - [Ext] swapExactTokensForETH #
  - [Ext] swapETHForExactTokens ($)
  - [Ext] quote
  - [Ext] getAmountOut
  - [Ext] getAmountIn
  - [Ext] getAmountsOut
  - [Ext] getAmountsIn

+ [Int] IUniswapV2Router02 (IUniswapV2Router01)

- [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
- [Ext] removeLiquidityETHWithPermitSupportingFeeOnTransferTokens #
- [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
- [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)
- [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #

+ [Lib] TransferHelper
- [Int] safeApprove #
- [Int] safeTransfer #
- [Int] safeTransferFrom #
- [Int] safeTransferETH #

+ DOGOToken (ERC20, Ownable)
- [Pub] <Constructor> #
  - modifiers: ERC20
- [Ext] <Fallback> ($)
- [Ext] createSwapPair #
  - modifiers: onlyOwner
- [Pub] updateDividendTracker #
  - modifiers: onlyOwner
- [Pub] updateUniswapV2Router #
  - modifiers: onlyOwner
- [Pub] excludeFromFees #
  - modifiers: onlyOwner
- [Ext] excludeMultipleAccountsFromFees #
  - modifiers: onlyOwner
- [Ext] setMarketingWallet #
  - modifiers: onlyOwner
- [Ext] setDOGERewardsFee #
  - modifiers: onlyOwner
- [Ext] setLiquiditFee #
  - modifiers: onlyOwner
- [Ext] setMarketingFee #
  - modifiers: onlyOwner
- [Ext] setSwapTokensAtAmount #
  - modifiers: onlyOwner
- [Pub] setAutomatedMarketMakerPair #
  - modifiers: onlyOwner
- [Ext] blacklistAddress #
  - modifiers: onlyOwner
- [Prv] _setAutomatedMarketMakerPair #
- [Pub] updateGasForProcessing #
  - modifiers: onlyOwner
- [Ext] updateClaimWait #
  - modifiers: onlyOwner
- [Ext] getClaimWait
- [Ext] getTotalDividendsDistributed
- [Pub] isExcludedFromFees
- [Pub] withdrawableDividendOf
- [Pub] dividendTokenBalanceOf
- [Ext] excludeFromDividends #
  - modifiers: onlyOwner
- [Ext] getAccountDividendsInfo
- [Ext] getAccountDividendsInfoAtIndex
- [Ext] processDividendTracker #

  - [Ext] claim #
  - [Ext] getLastProcessedIndex
  - [Ext] getNumberOfDividendTokenHolders
  - [Int] _transfer #
  - [Prv] swapAndSendToFee #
  - [Prv] swapAndLiquify #
  - [Prv] swapTokensForEth #
  - [Prv] swapTokensForDoge #
  - [Prv] addLiquidity #
  - [Prv] swapAndSendDividends #

+ DOGODividendTracker (Ownable, DividendPayingToken)
  - [Pub] <Constructor> #
    - modifiers: DividendPayingToken
  - [Int] _transfer #
  - [Pub] withdrawDividend #
  - [Ext] excludeFromDividends #
    - modifiers: onlyOwner
  - [Ext] updateClaimWait #
    - modifiers: onlyOwner
  - [Ext] getLastProcessedIndex
  - [Ext] getNumberOfTokenHolders
  - [Pub] getAccount
  - [Pub] getAccountAtIndex
  - [Prv] canAutoClaim
  - [Ext] setBalance #
    - modifiers: onlyOwner
  - [Pub] process #
  - [Pub] processAccount #
    - modifiers: onlyOwner

($) = payable function
= non-constant function

Issues Checking Status

	Issue description	Checking status
1.	Compiler errors.	Passed
2.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3.	Possible delays in data delivery.	Passed
4.	Oracle calls.	Passed
5.	Front running.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow.	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Low issue
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	Passed
12.	The impact of the exchange rate on the logic.	Passed
13.	Private user data leaks.	Passed
14.	Malicious Event log.	Passed
15.	Scoping and Declarations.	Passed
16.	Uninitialized storage pointers.	Passed
17.	Arithmetic accuracy.	Passed
18.	Design Logic.	Passed
19.	Cross-function race conditions.	Passed
20.	Safe Open Zeppelin contracts implementation and usage.	Passed
21.	Fallback function security.	Passed

Security Issues

High Severity Issues

No high severity issues found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

1. Out of gas

Issue:

- The function `excludeMultipleAccountsFromFees()` uses the loop to exclude multiple accounts from fees. Function will be aborted with `OUT_OF_GAS` exception if there will be a long addresses list.

Notes:

- Dividend tracker may be changed. So that logic of setBalance and other functions could be another and not audited.

Owner privileges (In the period when the owner is not renounced)

- Owner can update swap pair.
- Owner can change dividend tracker.
- Owner can change Uniswap router address.
- Owner can exclude from the fees.
- Owner can change marketing wallets.
- Owner can change DOGE rewards, liquidity and marketings fee.
- Owner can change minimum tokens amount to swap.
- Owner can exclude and include addresses in automatedMarketMakerPairs array.
- Owner can blacklist addresses.
- Owner can change gas for processing.
- Owner can update claimWait value.
- Owner can exclude from dividends.

Conclusion

Smart contracts do not contain high severity issues! Liquidity pair contract's security is not checked due to out of scope. The further transfers and operations with the funds raise are not related to this particular contract.

Liquidity locking details provided by the team:
https://bscscan.com/tx/0xa9589834521d745152afe9a495ca342802ce25f237d178742c7dc8412c873480

TechRate note:

Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.