Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

December, 2021

Audit Details

Audited project

DOGEDI

Deployer address

0xc173fca91287e03dd346b1959c3ff617f627bf68

Client contacts:

DOGEDI team

Blockchain

Binance Smart Chain

Project website:

Not provided

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 DOGEDI to perform an audit of smart contracts:

https://bscscan.com/address/0xdc49d53330317cbc6924fa53042e0c9bca0a8d63#code%23L1

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

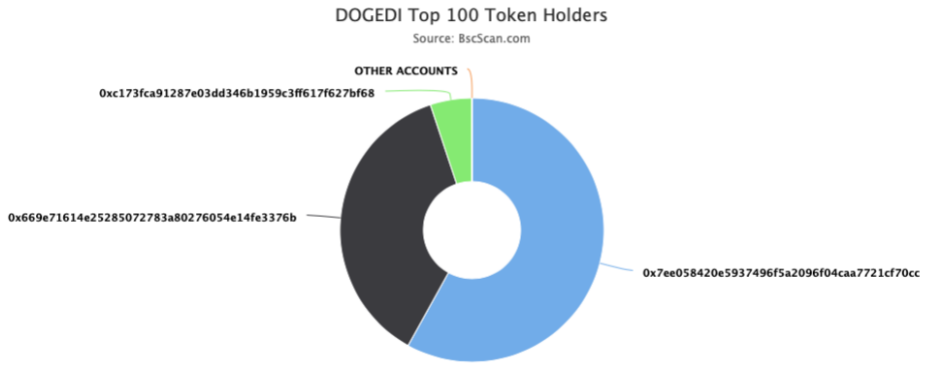
Token contract details for 01.12.2021

Contract name	DOGEDI
Contract address	0xDc49d53330317cBc6924fA53042e0C9bCa0A8d63
Total supply	1,000,000,000,000,000
Token ticker	DOGEDI
Decimals	12
Token holders	3
Transactions count	4
Top 100 holders dominance	100.00%
Dividend tracker	0x64bd20b5bcbee34b3c3619c295b54131a7548543
Total fees	15
BNB rewards fee	4
Uniswap V2 pair	0xd2cd4fa510731907c579646e1df7c7afe7155ccd
Contract deployer address	0xc173fca91287e03dd346b1959c3ff617f627bf68
Contract's current owner address	0xc173fca91287e03dd346b1959c3ff617f627bf68

DOGEDI Token Distribution

The top 100 holders collectively own 100.00% (1,000,000,000,000,000.00 Tokens) of DOGEDI

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 3

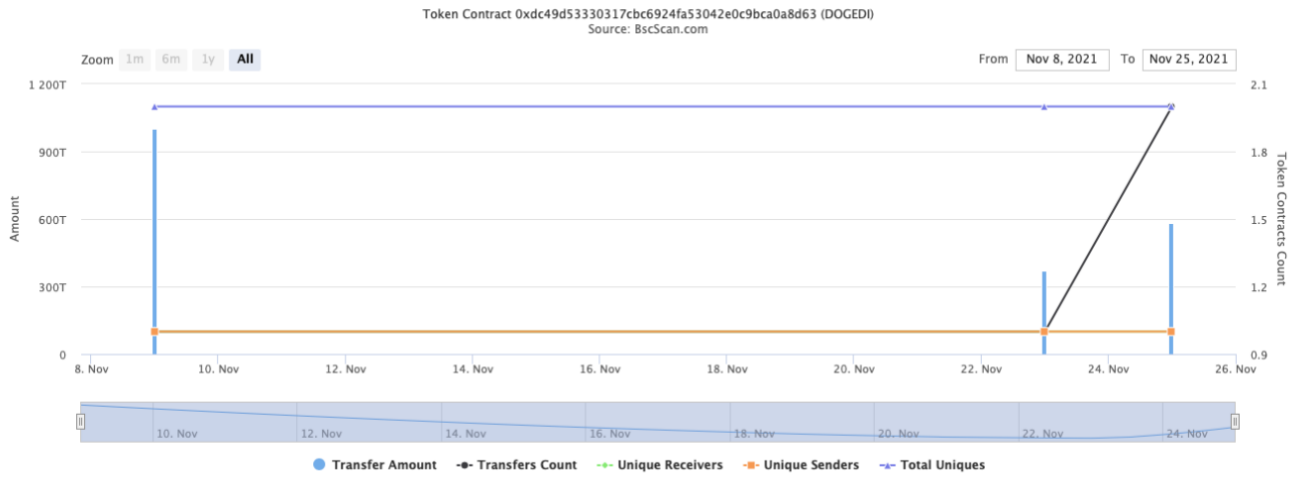


(A total of 1,000,000,000,000,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

DOGEDI Contract Interaction Details

Time Series: Token Contract Overview

Tue 9, Nov 2021 - Thu 25, Nov 2021



DOGEDI Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x7ee058420e5937496f5a2096f04caa7721cf70cc	580,585,000,000,000	58.0585%
2	0x669e71614e25285072783a80276054e14fe3376b	368,415,000,000,000	36.8415%
3	0xc173fca91287e03dd346b1959c3ff617f627bf68	51,000,000,000,000	5.1000%

Contract functions details

+ [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 #

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

+ [Lib] SignedSafeMath
  - [Int] mul
  - [Int] div
  - [Int] sub
  - [Int] add

+ [Lib] SafeMath
  - [Int] tryAdd
  - [Int] trySub
  - [Int] tryMul
  - [Int] tryDiv
  - [Int] tryMod
  - [Int] add

- [Int] sub
- [Int] mul
- [Int] div
- [Int] mod
- [Int] sub
- [Int] div
- [Int] mod

+ [Lib] SafeCast
- [Int] toUint224
- [Int] toUint128
- [Int] toUint96
- [Int] toUint64
- [Int] toUint32
- [Int] toUint16
- [Int] toUint8
- [Int] toUint256
- [Int] toInt128
- [Int] toInt64
- [Int] toInt32
- [Int] toInt16
- [Int] toInt8
- [Int] toInt256

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

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

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

+ 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 #
- [Int] _afterTokenTransfer #

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

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

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

+ [Int] DividendPayingTokenInterface
- [Ext] dividendOf
- [Ext] distributeDividends ($)
- [Ext] withdrawDividend #

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

+ DOGEDIDividendTracker (DividendPayingToken, Ownable)
- [Pub] <Constructor> #
  - modifiers: DividendPayingToken
- [Int] _transfer
- [Pub] withdrawDividend

  - [Ext] excludeFromDividends #
    - modifiers: onlyOwner
  - [Ext] updateClaimWait #
    - modifiers: onlyOwner
  - [Ext] getLastProcessedIndex
  - [Ext] getNumberOfTokenHolders
  - [Pub] setMinimumTokenBalanceForDividends #
    - modifiers: onlyOwner
  - [Pub] getAccount
  - [Pub] getAccountAtIndex
  - [Prv] canAutoClaim
  - [Ext] setBalance #
    - modifiers: onlyOwner
  - [Pub] process #
  - [Pub] processAccount #
    - modifiers: onlyOwner

+ SafeToken (Ownable)
  - [Pub] <Constructor> #
  - [Pub] setSafeManager #
    - modifiers: onlyOwner
  - [Ext] withdraw #
  - [Ext] withdrawBNB #

+ LockToken (Ownable)
  - [Pub] <Constructor> #
  - [Ext] openTrade #
    - modifiers: onlyOwner
  - [Ext] includeToWhiteList #
    - modifiers: onlyOwner

+ DOGEDI (ERC20, Ownable, SafeToken, LockToken)
  - [Pub] setFee #
    - modifiers: onlyOwner
  - [Pub] prepareForPresale #
    - modifiers: onlyOwner
  - [Pub] afterPresale #
    - modifiers: onlyOwner
  - [Pub] setExtraFeeOnSell #
    - modifiers: onlyOwner
  - [Pub] <Constructor> #
    - modifiers: ERC20
  - [Ext] <Fallback> ($)
  - [Pub] updateUniswapV2Router #
    - modifiers: onlyOwner
  - [Pub] excludeFromFees #
    - modifiers: onlyOwner
  - [Pub] setExcludeFromMaxTx #
    - modifiers: onlyOwner
  - [Pub] setExcludeFromAll #
    - modifiers: onlyOwner
  - [Pub] excludeMultipleAccountsFromFees #
    - modifiers: onlyOwner
  - [Pub] setAutomatedMarketMakerPair #
    - modifiers: onlyOwner

- [Prv] _setAutomatedMarketMakerPair #
- [Pub] updateGasForProcessing #
  - modifiers: onlyOwner
- [Ext] updateClaimWait #
  - modifiers: onlyOwner
- [Ext] getClaimWait
- [Ext] getTotalDividendsDistributed
- [Pub] isExcludedFromFees
- [Pub] isExcludedFromMaxTx
- [Pub] withdrawableDividendOf
- [Pub] dividendTokenBalanceOf
- [Ext] getAccountDividendsInfo
- [Ext] getAccountDividendsInfoAtIndex
- [Ext] processDividendTracker #
- [Ext] claim #
- [Ext] getLastProcessedIndex
- [Ext] getNumberOfDividendTokenHolders
- [Ext] excludeFromDividends #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Pub] setMarketingWallet #
  - modifiers: onlyOwner
- [Pub] setBuybackWallet #
  - modifiers: onlyOwner
- [Pub] setSwapTokensAtAmount #
  - modifiers: onlyOwner
- [Pub] setMaxSellTransactionAmount #
  - modifiers: onlyOwner
- [Int] _transfer #
  - modifiers: open
- [Pub] manualBurn #
  - modifiers: onlyOwner
- [Prv] swapAndLiquify #
  - modifiers: lockTheSwap
- [Prv] swapTokensForBnb #
- [Pub] setPrivateTokenPrice #
  - modifiers: onlyOwner
- [Pub] setPrivateSaleEnabled #
  - modifiers: onlyOwner
- [Pub] buyToken ($)
- [Pub] withdrawPrivateSale #
  - modifiers: onlyOwner
- [Pub] includeToPrivateWhiteList #
  - modifiers: onlyOwner
- [Pub] getWhiteListPrivateAddress
- [Pub] getWhiteListPrivate

($) = 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 issues
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.	Low issues
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. Rounding error

Issue:

- At each calculation with division, it is goes first. In Solidity we don't have floating points, but instead we get rounding errors.

```
function swapAndLiquify(uint256 contractTokenBalance⬆) private lockTheSwap {
    // capture the contract's current ETH balance.
    // this is so that we can capture exactly the amount of ETH that the
    // swap creates, and not make the liquidity event include any ETH that
    // has been manually sent to the contract
    uint256 initialBalance = address(this).balance;

    // swap tokens for ETH
    swapTokensForBnb(contractTokenBalance⬆, address(this)); // <- this breaks the ETH -> HATE swap when swap+liquify is triggered

    uint256 deltaBalance = address(this).balance-initialBalance;

    uint256 marketingAmount = deltaBalance.div(totalFees).mul(marketingFee);
    marketingWallet.transfer(marketingAmount);

    uint256 buybackAmount = deltaBalance.div(totalFees).mul(buybackFee);
    buybackWallet.transfer(buybackAmount);

    uint256 dividends = address(this).balance;
    (bool success,) = address(dividendTracker).call{value: dividends}("");

    if(success) {
        emit SendDividends(contractTokenBalance⬆, dividends);
    }
}
```

Recommendation:

Do division after multiplication.

2. 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.

```
function excludeMultipleAccountsFromFees(address[] calldata accounts⬆, bool excluded⬆) public onlyOwner {
    for(uint256 i = 0; i < accounts⬆.length; i++)
    {
        _isExcludedFromFees[accounts⬆[i]] = excluded⬆;
    }
    emit ExcludeMultipleAccountsFromFees(accounts⬆, excluded⬆);
}
```

- The function `includeToWhiteList()` also uses the loop to whitelist addresses. Function will be aborted with `OUT_OF_GAS` exception if there will be a long addresses list.

```
function includeToWhiteList(address[] memory _users) external onlyOwner {
    for(uint8 i = 0; i < _users.length; i++) {
        _whiteList[_users[i]] = true;
    }
}
```

- The function `includeToPrivateWhiteList()` also uses the loop to whitelist private addresses. Function will be aborted with `OUT_OF_GAS` exception if there will be a long addresses list.

```
function includeToPrivateWhiteList(address[] memory _users⬆) public onlyOwner {
    for(uint8 i = 0; i < _users⬆.length; i++) {
        whiteListPrivate[_users⬆[i]] = true;
        whiteListPrivateList.push(_users⬆[i]);
    }
}
```

Recommendation:
Check that the addresses array length is not too big.

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

- Owner can change fees.
- Owner can run presale and after presale modes.
- Owner can change extraFeeOnSell value.
- Owner can change Uniswap router.
- Owner can exclude from fees.
- Owner can exclude from max transaction amount.
- Owner can exclude from and include in addresses to automatedMarketMakerPairs array.
- Owner can change gas for processing.
- Owner can change claimWait value.
- Owner can exclude from dividends.
- Owner can change marketing and buyback wallet addresses.
- Owner can change swapTokensAtAmount value.
- Owner can call manual burn function.
- Owner can change pricePrivate value.
- Owner can enable/disable privateSaleIsActive.
- Owner can withdraw contract BNBs.

Conclusion

Smart contracts contain low 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 NOT provided by the team.

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.