



TechRate

AUDIT COMPANY

Smart Contract Security Audit

Audit Details



Audited project
BananaCake



Deployer address
0x241f05654d6327ab6d66befbd59a08fde20abdf7



Client contacts:
BananaCake team



Blockchain
Binance Smart Chain



Project website:
<https://www.banana-cake.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 BananaCake to perform an audit of smart contracts:

<https://bscscan.com/address/0xC65d0B763E7E7EDD0Cfc3800192300a7c6F338f9#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 08.09.2021

Contract name	BananaCake
Contract address	0xC65d0B763E7E7EDD0Cfc3800192300a7c6F338f9
Total supply	100,000,000,000
Token ticker	BANACAKE
Decimals	18
Token holders	1
Transactions count	1
Top 100 holders dominance	100.00%
Cake/Banana Dividend token	0x0e09fabb73bd3ade0a17ecc321fd13a19e81ce82 0x603c7f932ed1fc6575303d8fb018fdccb0f39a95
Total fees	17
Dividend rewards fee	5
Uniswap V2 pair	0xc25288fd68bf02eab2aba0b88b61b3673735553f
Contract deployer address	0x241f05654d6327ab6d66befbd59a08fde20abdf7
Contract's current owner address	0x241f05654d6327ab6d66befbd59a08fde20abdf7

BananaCake Token Distribution

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

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 1

BananaCake Top 100 Token Holders
Source: BscScan.com



(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)

BananaCake Contract Interaction Details

Time Series: Token Contract Overview

Mon 6, Sept 2021 - Mon 6, Sept 2021

Token Contract 0xC65d08763E7E7EDD0Cfc3800192300a7c6F338F9 (BananaCake)
Source: BscScan.com



BananaCake Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x241f05654d6327ab6d66befbd59a08fde20abdf7	100,000,000,000	100.0000%



Contract functions details

- + Context
 - [Int] _msgSender
 - [Int] _msgData
- + Ownable (Context)
 - [Pub] <Constructor> #
 - [Pub] owner
 - [Pub] renounceOwnership #
 - modifiers: onlyOwner
 - [Pub] transferOwnership #
 - modifiers: onlyOwner
 - [Prv] _setOwner #
- + [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 #
- + [Int] IUniswapV2Factory
 - [Ext] feeTo
 - [Ext] feeToSetter
 - [Ext] getPair
 - [Ext] allPairs

- [Ext] allPairsLength
 - [Ext] createPair #
 - [Ext] setFeeTo #
 - [Ext] setFeeToSetter #
- + [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] 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] SafeMathInt
 - [Int] mul
 - [Int] div
 - [Int] sub
 - [Int] add
 - [Int] toUint256Safe

- + [Lib] SafeMathUint
 - [Int] toInt256Safe

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

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

- + DividendPayingToken (ERC20, IDividendPayingToken, IDividendPayingTokenOptional)
 - [Pub] <Constructor> #
 - modifiers: ERC20
 - [Ext] <Fallback> (\$)
 - [Pub] distributeDividends (\$)
 - [Pub] distributeDividends #
 - [Pub] withdrawDividend #
 - [Ext] setDividendTokenAddress #
 - [Ext] setMinTokenBeforeSendDividend #
 - [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 #

- + CakeDividendTracker (DividendPayingToken, Ownable)
 - [Pub] <Constructor> #
 - modifiers: DividendPayingToken
 - [Int] _transfer
 - [Pub] withdrawDividend
 - [Ext] setDividendTokenAddress #
 - modifiers: onlyOwner
 - [Ext] updateMinimumTokenBalanceForDividends #
 - modifiers: onlyOwner
 - [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
- + BananaDividendTracker (DividendPayingToken, Ownable)
- [Pub] <Constructor> #
 - modifiers: DividendPayingToken
 - [Int] _transfer
 - [Pub] withdrawDividend
 - [Ext] setDividendTokenAddress #
 - modifiers: onlyOwner
 - [Ext] updateMinimumTokenBalanceForDividends #
 - modifiers: onlyOwner
 - [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
- + BananaCake (ERC20, Ownable)
- [Pub] <Constructor> #
 - modifiers: ERC20
 - [Ext] <Fallback> (\$)
 - [Ext] prepareForPartnerOrExchangeListing #
 - modifiers: onlyOwner
 - [Ext] setMaxBuyTransaction #
 - modifiers: onlyOwner
 - [Ext] setMaxSellTransaction #
 - modifiers: onlyOwner
 - [Ext] updateBananaDividendToken #
 - modifiers: onlyOwner
 - [Ext] updateMinBananaBeforeSendDividend #
 - modifiers: onlyOwner
 - [Ext] updateCakeDividendToken #
 - modifiers: onlyOwner
 - [Ext] updateMinCakeBeforeSendDividend #
 - modifiers: onlyOwner
 - [Ext] getMinBananaBeforeSendDividend
 - [Ext] getMinCakeBeforeSendDividend
 - [Ext] setSendBananaInTx #

- modifiers: onlyOwner
- [Ext] setSendCakeInTx #
 - modifiers: onlyOwner
- [Ext] setCakeDividendPriority #
 - modifiers: onlyOwner
- [Ext] updateTeamWallet #
 - modifiers: onlyOwner
- [Ext] updateMarketingWallet #
 - modifiers: onlyOwner
- [Ext] setMaxWalletToken #
 - modifiers: onlyOwner
- [Ext] setSwapTokensAtAmount #
 - modifiers: onlyOwner
- [Ext] setSellTransactionMultiplier #
 - modifiers: onlyOwner
- [Ext] afterPreSale #
 - modifiers: onlyOwner
- [Ext] setTradingIsEnabled #
 - modifiers: onlyOwner
- [Ext] setBuyBackMode #
 - modifiers: onlyOwner
- [Pub] setMinimumBalanceRequired #
 - modifiers: onlyOwner
- [Pub] setMinimumSellOrderAmount #
 - modifiers: onlyOwner
- [Ext] setBuyBackUpperLimit #
 - modifiers: onlyOwner
- [Ext] setBuyBackAndLiquifyEnabled #
 - modifiers: onlyOwner
- [Ext] setCakeDividendEnabled #
 - modifiers: onlyOwner
- [Ext] setBananaDividendEnabled #
 - modifiers: onlyOwner
- [Ext] setMarketingEnabled #
 - modifiers: onlyOwner
- [Ext] updateCakeDividendTracker #
 - modifiers: onlyOwner
- [Ext] updateBananaDividendTracker #
 - modifiers: onlyOwner
- [Ext] updateCakeDividendRewardFee #
 - modifiers: onlyOwner
- [Ext] updateBananaDividendRewardFee #
 - modifiers: onlyOwner
- [Ext] updateMarketingFee #
 - modifiers: onlyOwner
- [Ext] updateBuyBackAndLiquidityFee #
 - modifiers: onlyOwner
- [Ext] updateUniswapV2Router #
 - modifiers: onlyOwner
- [Pub] excludeFromFees #
 - modifiers: onlyOwner
- [Pub] excludeFromDividend #
 - modifiers: onlyOwner
- [Ext] excludeMultipleAccountsFromFees #
 - modifiers: onlyOwner

- [Pub] setAutomatedMarketMakerPair #
 - modifiers: onlyOwner
- [Prv] _setAutomatedMarketMakerPair #
 - modifiers: onlyOwner
- [Ext] updateGasForProcessing #
 - modifiers: onlyOwner
- [Ext] updateMinimumBalanceForDividends #
 - modifiers: onlyOwner
- [Ext] updateClaimWait #
 - modifiers: onlyOwner
- [Ext] getCakeClaimWait
- [Ext] getBananaClaimWait
- [Ext] getTotalCakeDividendsDistributed
- [Ext] getTotalBananaDividendsDistributed
- [Pub] getIsExcludedFromFees
- [Ext] withdrawableCakeDividendOf
- [Ext] withdrawableBananaDividendOf
- [Ext] cakeDividendTokenBalanceOf
- [Ext] bananaDividendTokenBalanceOf
- [Ext] getAccountCakeDividendsInfo
- [Ext] getAccountBananaDividendsInfo
- [Ext] getAccountCakeDividendsInfoAtIndex
- [Ext] getAccountBananaDividendsInfoAtIndex
- [Ext] processDividendTracker #
 - modifiers: onlyOwner
- [Int] rand
- [Ext] claim #
- [Ext] getLastCakeDividendProcessedIndex
- [Ext] getLastBananaDividendProcessedIndex
- [Ext] getNumberOfCakeDividendTokenHolders
- [Ext] getNumberOfBananaDividendTokenHolders
- [Int] _transfer #
- [Prv] swapAndLiquify #
- [Prv] addLiquidity #
- [Prv] buyBackAndBurn #
- [Pub] manualBuyBackAndBurn #
 - modifiers: onlyOwner
- [Prv] swapTokensForBNB #
- [Prv] swapTokensForDividendToken #
- [Prv] swapAndSendCakeDividends #
- [Prv] swapAndSendBananaDividends #
- [Prv] transferToWallet #
- [Prv] transferDividends #

(\$) = 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.

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.

Recommendation:

Be careful about accounts array length.

Notes:

- Owner can change dividend tracker that could be not audited and some functions may work in different ways.
- DividendPayingToken in the contract has wrong addresses checking to run contract functions. But in actual Cake/Banana Dividend tracker addresses for key functions are right.

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

- Owner can whitelist addresses (exclude from fees and dividends).
- Owner can change max buy and sell transaction amounts and fees.
- Owner can change dividend tokens addresses.
- Owner can change sendBananaInTx and sendCakeInTx values.
- Owner can change Cake dividends priority.
- Owner can change min value to send dividends.
- Owner can change marketing address.
- Owner can change max wallet token number.
- Owner can change swapTokensAtAmount.
- Owner can change sell transaction multiplier.
- Owner can enable after presale mode.
- Owner can enable and disable trading.
- Owner can enable buyback mode.
- Owner can change minimumBalanceRequired, minimumSellOrderAmount, buyBackUpperLimit values.
- Owner can enable and disable buyback and liquify.
- Owner can enable and disable CAKE and Banana dividends.
- Owner can enable and disable marketing fee.
- Owner can change CAKE and Banana dividendTrackers.
- Owner can change CAKE and Banana dividends rewards fee, marketing buyback and liquidity fee.
- Owner can change Uniswap router.
- Owner can exclude from fees and dividends.
- Owner can exclude and include addresses in automatedMarketMakerPairs array.
- Owner can change gas for processing.
- Owner can change minimum token balance for dividends.
- Owner can update claimWait value.
- Owner can manually buyback and burn.
- 0x241F05654d6327aB6d66bEfBD59A08fDe20abdf7 address can setDividendTokenAddress and setMinTokenBeforeSendDividend.

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 are 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.