DiamondHold

Smart Contract Security Audit

May, 2021

TechRate

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 DiamondHold to perform an audit of smart contracts:

- *https://bscscan.com/address/0xeE8feAeE52CE378BA356A5772BBa29d08AF25cdB#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.

Issues Checking Status

№	Issue description.	Checking status
1	Compiler warnings.	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.	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 Zeppelin module.	Passed
21	Fallback function security.	Passed

Functions outline

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

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

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

+ [Lib] Address
  - [Int] isContract
  - [Int] sendValue #
  - [Int] functionCall #
  - [Int] functionCall #
  - [Int] functionCallWithValue #
  - [Int] functionCallWithValue #
  - [Prv] _functionCallWithValue #

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

+ [Int] IUniswapV2Factory

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

+ [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] 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 #

+ DiamondHold (Context, IERC20, Ownable)
  - [Pub] <Constructor> #
  - [Ext] <Fallback> ($)
  - [Pub] withdraw #
    - modifiers: onlyOwner
  - [Pub] withdrawToken #
    - modifiers: onlyOwner
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals
  - [Pub] allowance
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Pub] approve #
  - [Pub] transfer #
  - [Pub] transferFrom #
  - [Ext] totalSupply
  - [Ext] totalFees
  - [Ext] getTierOneTotal
  - [Ext] getTierTwoTotal
  - [Ext] getTierThreeTotal
  - [Ext] getTierFourTotal
  - [Pub] balanceOf
  - [Pub] getBalanceOf
  - [Int] getTokenBalanceFromReflection
  - [Pub] getWalletAge
  - [Pub] getWalletTier
  - [Pub] getWalletReflectionFactor

- [Pub] setWalletTier #
  - modifiers: onlyOwner
- [Pub] isExcludedFromReflection
- [Pub] excludeFromReflection #
  - modifiers: onlyOwner
- [Ext] includeInReflection #
  - modifiers: onlyOwner
- [Prv] removeAllFees #
- [Prv] restoreAllFees #
- [Ext] isExcludedFromFee
- [Ext] excludeFromFees #
  - modifiers: onlyOwner
- [Ext] includeInFees #
  - modifiers: onlyOwner
- [Ext] disableStartOnTier2 #
  - modifiers: onlyOwner
- [Ext] isExchange
- [Ext] enableExchange #
  - modifiers: onlyOwner
- [Ext] disableExchange #
  - modifiers: onlyOwner
- [Ext] setTierOneReflectionTaxPercent #
  - modifiers: onlyOwner
- [Ext] setTierTwoReflectionTaxPercent #
  - modifiers: onlyOwner
- [Ext] setTierThreeReflectionTaxPercent #
  - modifiers: onlyOwner
- [Ext] setTierFourReflectionTaxPercent #
  - modifiers: onlyOwner
- [Ext] setLiquidityTaxPercent #
  - modifiers: onlyOwner
- [Ext] setMaxTokenPercent #
  - modifiers: onlyOwner
- [Ext] setNumTokensLiquidity #
  - modifiers: onlyOwner
- [Pub] setAutoLiquidityProtocol #
  - modifiers: onlyOwner
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _getTokenValues
- [Prv] _getReflectionValues
- [Prv] _reflectFee #
- [Prv] _takeLiquidity #
- [Prv] _approve #
- [Prv] _transfer #
- [Prv] swapAndLiquify #
  - modifiers: lockTheSwap

- [Prv] swapTokensForBNB #
- [Prv] addLiquidity #
- [Prv] _tokenTransfer #

($) = payable function
= non-constant function

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 `includeInReflection` uses the loop to find and remove addresses from the `_excluded` list. Function will be aborted with `OUT_OF_GAS` exception if there will be a long excluded addresses list.
- The function `_getCurrentSupply` also uses the loop for evaluating total supply. It also could be aborted with `OUT_OF_GAS` exception if there will be a long excluded addresses list.

Recommendation:

Check that _excluded list will not be too big.

Owner privileges

- Owner can change the liquidity fee.
- Owner can change the maximum transaction amount.
- Owner can exclude from the fee.
- Owner can change the wallet's tier.
- Owner can change the tiers tax fee.

Conclusion

Smart contracts contain low severity issues and owner privileges. LP pair contract security is not checked due to out of scope.

No liquidity locking info provided by the team!

Liquidity will be added to the owner's wallet and he should lock it himself.

```
function addLiquidity(uint256 tokenAmount⬆, uint256 bnbAmount⬆) private {
    // Approve token transfer to cover all possible scenarios
    _approve(address(this), address(pancakeswapV2Router), tokenAmount⬆);

    // Adds the liquidity and gives the LP tokens to the owner of this contract
    // The LP tokens need to be manually locked
    pancakeswapV2Router.addLiquidityETH{value: bnbAmount⬆}(
        address(this),
        tokenAmount⬆,
        0, // slippage is unavoidable
        0, // slippage is unavoidable
        owner(),
        block.timestamp
    );
}
```

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.