Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details

Audited project

PokeDX

Deployer address

0xcbf4daedf93a623a6a82466c5ba1257a5fb0ca51

Client contacts:

PokeDX team

Blockchain

Binance Smart Chain

Project website:

https://pokedx.app

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 PokeDX to perform an audit of smart contracts:

https://bscscan.com/address/0x43a0c5eb1763a211aa3c05849a617f2ee0452767#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.

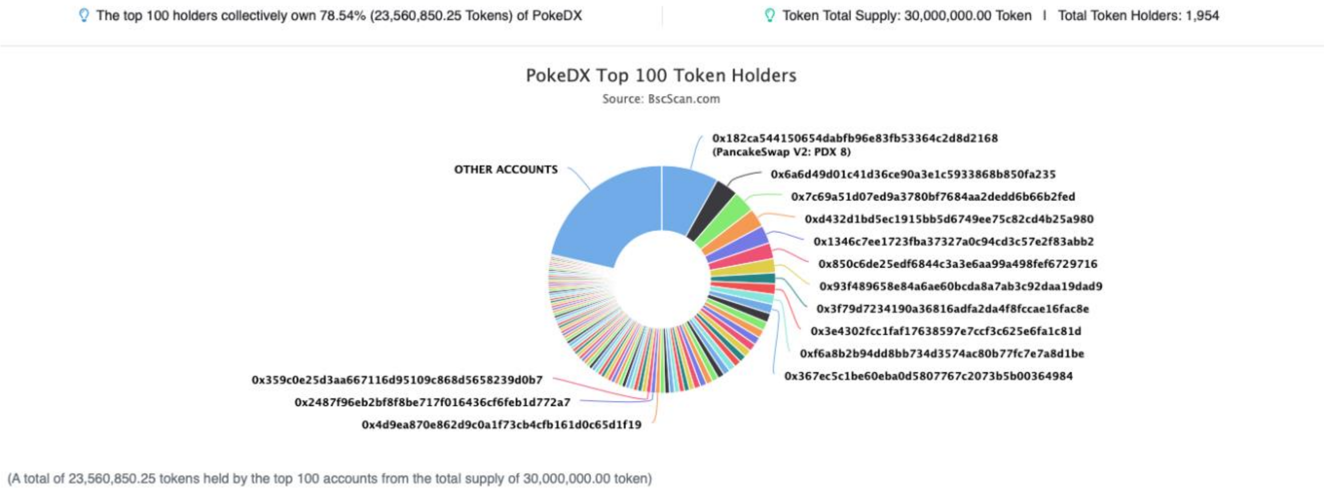
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 17.11.2021

Contract name	PokeDX
Contract address	0x43a0C5EB1763A211Aa3c05849A617f2eE0452767
Total supply	30,000,000
Token ticker	PDX
Decimals	9
Token holders	1,954
Transactions count	10,620
Top 100 holders dominance	78.54%
Sum of fees	40
LP Receiver	0xcbf4daedf93a623a6a82466c5ba1257a5fb0ca51
Contract deployer address	0xcbf4daedf93a623a6a82466c5ba1257a5fb0ca51
Contract’s current owner address	0xcbf4daedf93a623a6a82466c5ba1257a5fb0ca51

PokeDX Token Distribution



The top 100 holders collectively own 78.54% (23,560,850.25 Tokens) of PokeDX

Token Total Supply: 30,000,000.00 Token | Total Token Holders: 1,954

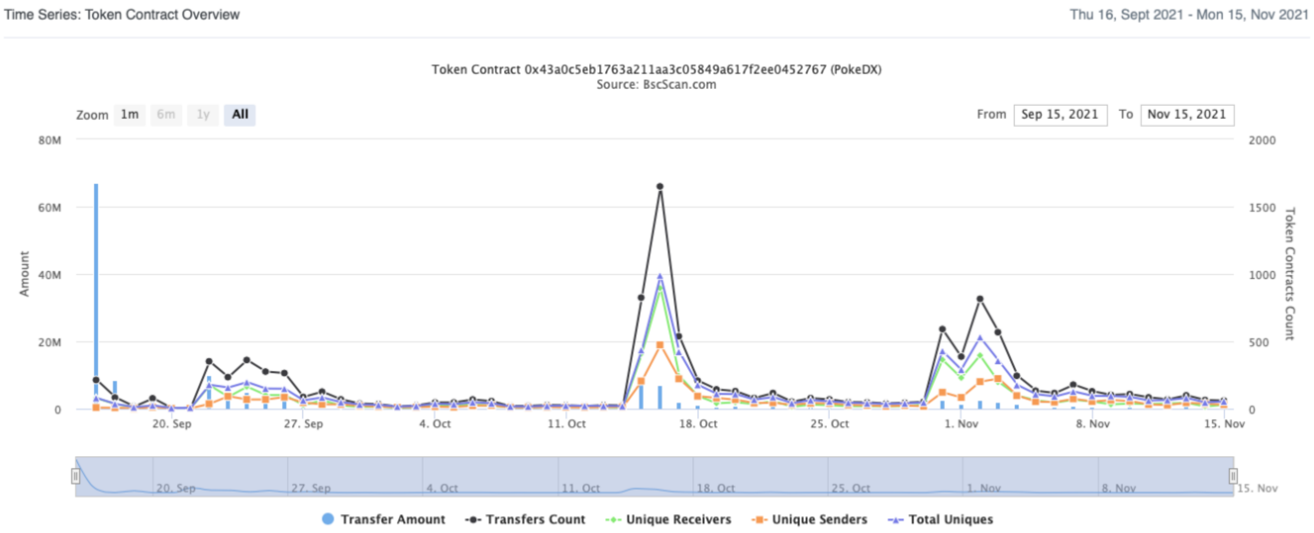
PokeDX Top 100 Token Holders

Source: BscScan.com

OTHER ACCOUNTS
0x182ca544150654dabfb96e83fb53364c2d8d2168
(PancakeSwap V2: PDX 8)
0x6a6d49d01c41d36ce90a3e1c5933868b850fa235
0x7c69a51d07ed9a3780bf7684aa2dedd6b66b2fed
0xd432d1bd5ec1915bb5d6749ee75c82cd4b25a980
0x1346c7ee1723fba37327a0c94cd3c57e2f83abb2
0x850c6de25edf6844c3a3e6aa99a498fef6729716
0x93f489658e84a6ae60bcda8a7ab3c92daa19dad9
0x3f79d7234190a36816adfa2da4f8fccae16fac8e
0x3e4302fcc1faf17638597e7ccf3c625e6fa1c81d
0xf6a8b2b94dd8bb734d3574ac80b77fc7e7a8d1be
0x367ec5c1be60eba0d5807767c2073b5b00364984
0x359c0e25d3aa667116d95109c868d5658239d0b7
0x2487f96eb2bf8f8be717f016436cf6feb1d772a7
0x4d9ea870e862d9c0a1f73cb4cfb161d0c65d1f19

(A total of 23,560,850.25 tokens held by the top 100 accounts from the total supply of 30,000,000.00 token)

PokeDX Contract Interaction Details



Time Series: Token Contract Overview

Thu 16, Sept 2021 - Mon 15, Nov 2021

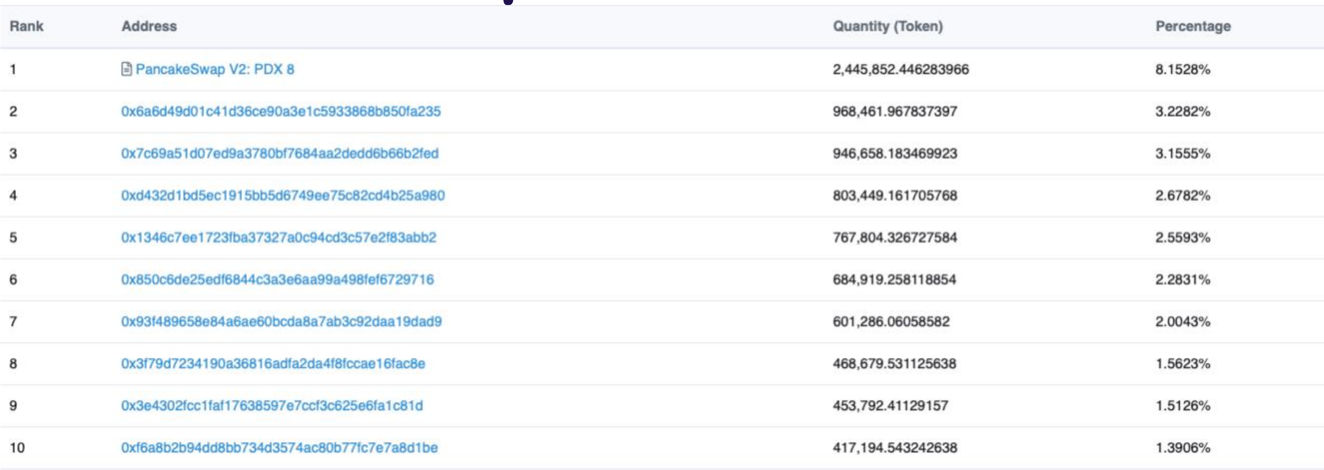
Token Contract 0x43a0c5eb1763a211aa3c05849a617f2ee0452767 (PokeDX)

Source: BscScan.com

Zoom 1m 6m 1y All

From Sep 15, 2021 To Nov 15, 2021

PokeDX Top 10 Token Holders



Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: PDX 8	2,445,852.446283966	8.1528%
2	0x6a6d49d01c41d36ce90a3e1c5933868b850fa235	968,461.967837397	3.2282%
3	0x7c69a51d07ed9a3780bf7684aa2dedd6b66b2fed	946,658.183469923	3.1555%
4	0xd432d1bd5ec1915bb5d6749ee75c82cd4b25a980	803,449.161705768	2.6782%
5	0x1346c7ee1723fba37327a0c94cd3c57e2f83abb2	767,804.326727584	2.5593%
6	0x850c6de25edf6844c3a3e6aa99a498fef6729716	684,919.258118854	2.2831%
7	0x93f489658e84a6ae60bcda8a7ab3c92daa19dad9	601,286.06058582	2.0043%
8	0x3f79d7234190a36816adfa2da4f8fccae16fac8e	468,679.531125638	1.5623%
9	0x3e4302fcc1faf17638597e7ccf3c625e6fa1c81d	453,792.41129157	1.5126%
10	0xf6a8b2b94dd8bb734d3574ac80b77fc7e7a8d1be	417,194.543242638	1.3906%

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] Address
  - [Int] isContract
  - [Int] sendValue #
  - [Int] functionCall #
  - [Int] functionCall #
  - [Int] functionCallWithValue #
  - [Int] functionCallWithValue #
  - [Int] functionStaticCall
  - [Int] functionStaticCall
  - [Int] functionDelegateCall #
  - [Int] functionDelegateCall #
  - [Prv] _verifyCallResult

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

+ [Int] IPancakeV2Factory
  - [Ext] createPair #

+ [Int] IPancakeV2Router
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidityETH ($)
  - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #
```

+ Pausable (Context)
  - [Pub] <Constructor> #
  - [Pub] paused
  - [Int] _pause #
    - modifiers: whenNotPaused
  - [Int] _unpause #
    - modifiers: whenPaused

+ Tokenomics
  - [Pub] <Constructor> #
  - [Prv] _addFee #
  - [Prv] _addFees #
  - [Int] _getFeesCount
  - [Prv] _getFeeStruct
  - [Int] _getFee
  - [Int] _addFeeCollectedAmount #
  - [Int] getCollectedFeeTotal

+ BaseRfiToken (IERC20, IERC20Metadata, Ownable, Pausable, Tokenomics)
  - [Pub] <Constructor> #
  - [Ext] name
  - [Ext] symbol
  - [Ext] decimals
  - [Ext] totalSupply
  - [Pub] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #
  - [Ext] burn #
  - [Int] _burnTokens #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Ext] isExcludedFromReward
  - [Ext] reflectionFromToken
  - [Int] tokenFromReflection
  - [Ext] excludeFromReward #
    - modifiers: onlyOwner
  - [Int] _exclude #
  - [Ext] includeInReward #
    - modifiers: onlyOwner
  - [Ext] setExcludedFromFee #
    - modifiers: onlyOwner
  - [Pub] isExcludedFromFee
  - [Int] _approve #
  - [Int] _isUnlimitedSender
  - [Int] _isUnlimitedRecipient
  - [Prv] _transfer #
  - [Prv] _transferTokens #
  - [Prv] _takeFees #
  - [Int] _getValues
  - [Int] _getCurrentRate
  - [Int] _getCurrentSupply
  - [Int] _beforeTokenTransfer #
  - [Int] _getSumOfFees

```
  - [Int] _isV2Pair
  - [Int] _redistribute #
  - [Int] _takeTransactionFees #
  - [Pub] pause #
    - modifiers: onlyOwner
  - [Pub] unpause #
    - modifiers: onlyOwner

+ Liquifier (Ownable)
  - [Ext] <Fallback> ($)
  - [Ext] _setNumberOfTokensToSwapToLiquidity #
    - modifiers: onlyOwner
  - [Ext] showNumberOfTokensToSwapToLiquidity
  - [Int] initializeLiquiditySwapper #
  - [Int] liquify #
  - [Prv] _setRouterAddress #
  - [Prv] _swapAndLiquify #
    - modifiers: lockTheSwap
  - [Prv] _swapTokensForEth #
  - [Ext] setLPReceiver #
    - modifiers: onlyOwner
  - [Ext] showLPReceiver
  - [Prv] _addLiquidity #
  - [Ext] setRouterAddress #
    - modifiers: onlyOwner
  - [Ext] setSwapAndLiquifyEnabled #
    - modifiers: onlyOwner
  - [Ext] withdrawLockedBNB #
    - modifiers: onlyOwner
  - [Int] _approveDelegate #

+ PokeDX (BaseRfiToken, Liquifier)
  - [Pub] <Constructor> #
  - [Int] _isV2Pair
  - [Int] _getSumOfFees
  - [Int] _beforeTokenTransfer #
  - [Int] _takeTransactionFees #
  - [Prv] _burn #
  - [Prv] _takeFee #
  - [Int] _approveDelegate #
  - [Ext] showFee
  - [Ext] increaseFee #
    - modifiers: onlyOwner
  - [Ext] decreaseFee #
    - 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 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.	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

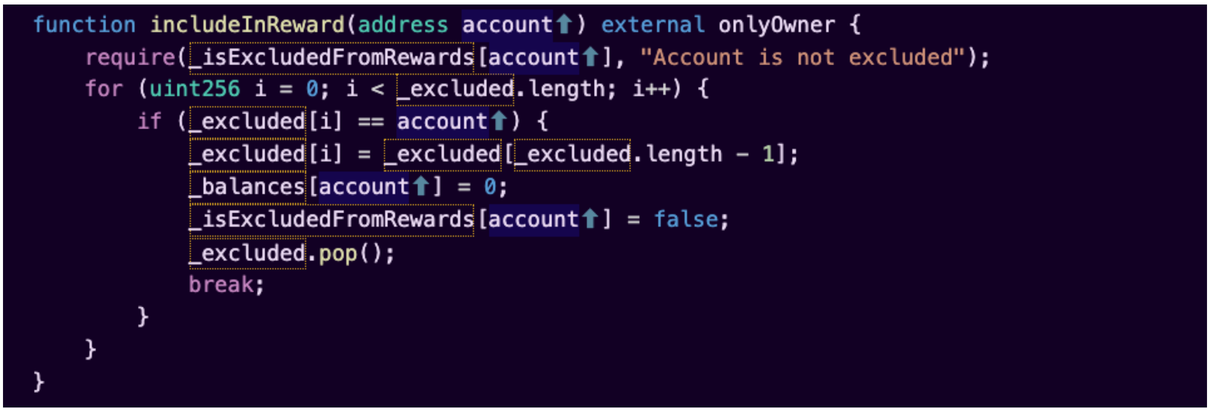
No medium severity issues found.

Low Severity Issues

1. Out of gas

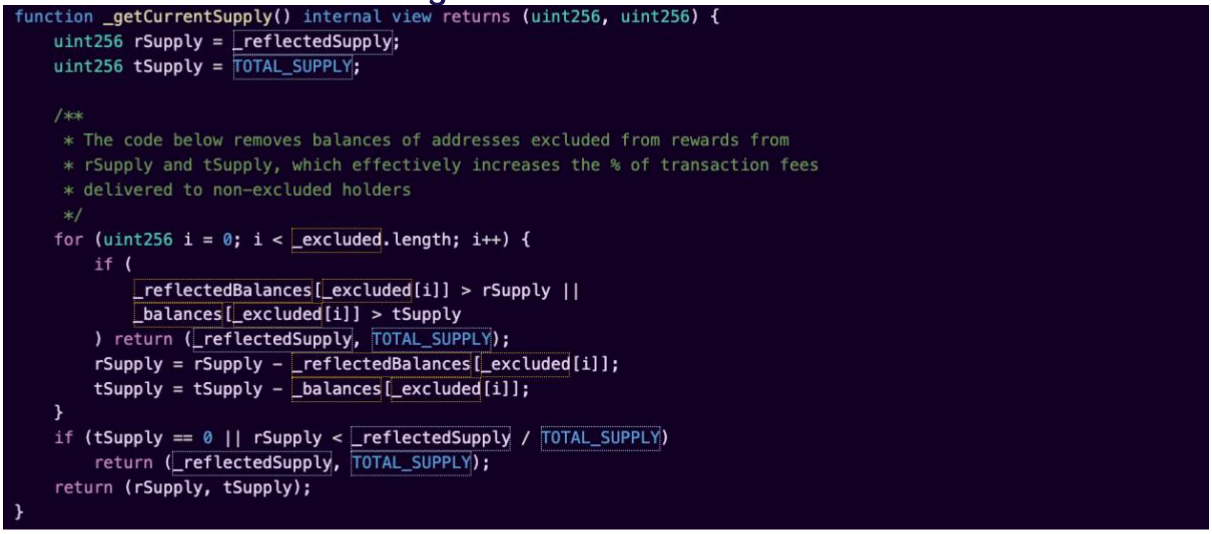
Issue:

- The function includeInReward() 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.



```
function includeInReward(address account) external onlyOwner {
    require(_isExcludedFromRewards[account], "Account is not excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _balances[account] = 0;
            _isExcludedFromRewards[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- 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.



```
function _getCurrentSupply() internal view returns (uint256, uint256) {
    uint256 rSupply = _reflectedSupply;
    uint256 tSupply = TOTAL_SUPPLY;

    /**
     * The code below removes balances of addresses excluded from rewards from
     * rSupply and tSupply, which effectively increases the % of transaction fees
     * delivered to non-excluded holders
     */
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _reflectedBalances[_excluded[i]] > rSupply ||
            _balances[_excluded[i]] > tSupply
        ) return (_reflectedSupply, TOTAL_SUPPLY);
        rSupply = rSupply - _reflectedBalances[_excluded[i]];
        tSupply = tSupply - _balances[_excluded[i]];
    }
    if (tSupply == 0 || rSupply < _reflectedSupply / TOTAL_SUPPLY)
        return (_reflectedSupply, TOTAL_SUPPLY);
    return (rSupply, tSupply);
}
```

Recommendation:

Check that the excluded array length is not too big.

Notes:

- The function _burnTokens() sends burn amount to burnAddress.

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

- Owner can pause/unpause contract.
- Owner can change numberOfTokensToSwapToLiquidity.
- Owner can change LPReceiver.
- Owner can exclude from the fee.
- Owner can increase/decrease fees.
- Owner can change router.
- Owner can withdraw contract BNBs.
- Owner can lock and unlock. By the way, using these functions the owner could retake privileges even after the ownership was renounced.

Conclusion

Smart contracts contain low severity issues! Liquidity pair contract's security is not checked due to out of scope.

Liquidity locking details provided by the team:
https://dxsale.app/app/v3/dxlockview?id=0&add=0xcBF4DAeDF93a623a6A82466C5ba1257A5FB0cA51&type=lplock&chain=BSC

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.