



# Smart Contract Security Audit

## Audit details:

Audited project:	BoggieCoin
Deployer address:	0x5b558F9454040A4C5b1Eff89d87ea26A8292bD31
Client contacts:	BoggieCoin team
Blockchain:	Binance Smart Chain
Project website:	<a href="https://boggiecoin.com">https://boggiecoin.com</a>

# 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 BoggieCoin to perform an audit of smart contracts:

- <https://bscscan.com/address/0x1570dD4DD36ad9E66D53179ecad33bF66dFD722b#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 29.05.2021.

Contract name:	BoggieCoin
Contract address:	0x1570dD4DD36ad9E66D53179ecad33bF66dFD722b
Total supply:	1975895879638290817189432
Token ticker:	BC
Decimals:	9
Token holders:	2,012
Transactions count:	4,420
Top 100 holders dominance:	82.37%
Liquidity fee:	12
Tax fee:	4
Total fees:	38153160219698633536231
Uniswap V2 pair:	0x47ef533877b3a8341c843d460cee4ed55455c017
Contract deployer address:	0x5b558F9454040A4C5b1Eff89d87ea26A8292bD31
Contract's current owner address:	0x5b558f9454040a4c5b1eff89d87ea26a8292bd31

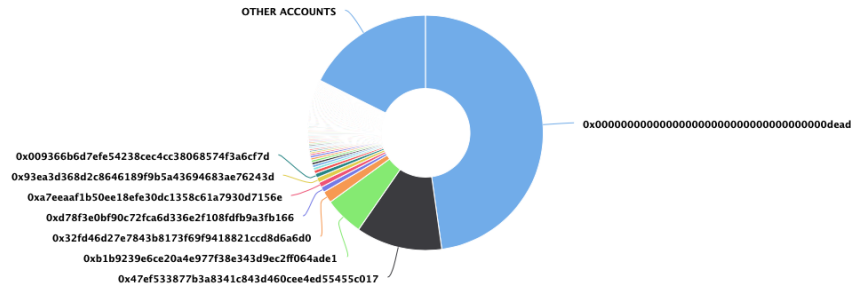
# BoggieCoin token distribution

The top 100 holders collectively own 82.37% (1,627,454,672,812,810.00 Tokens) of BoggieCoin

Token Total Supply: 1,975,895,879,638,290.82 Token | Total Token Holders: 2,012

BoggieCoin Top 100 Token Holders

Source: BscScan.com



(A total of 1,627,454,672,812,810.00 tokens held by the top 100 accounts from the total supply of 1,975,895,879,638,290.82 token)

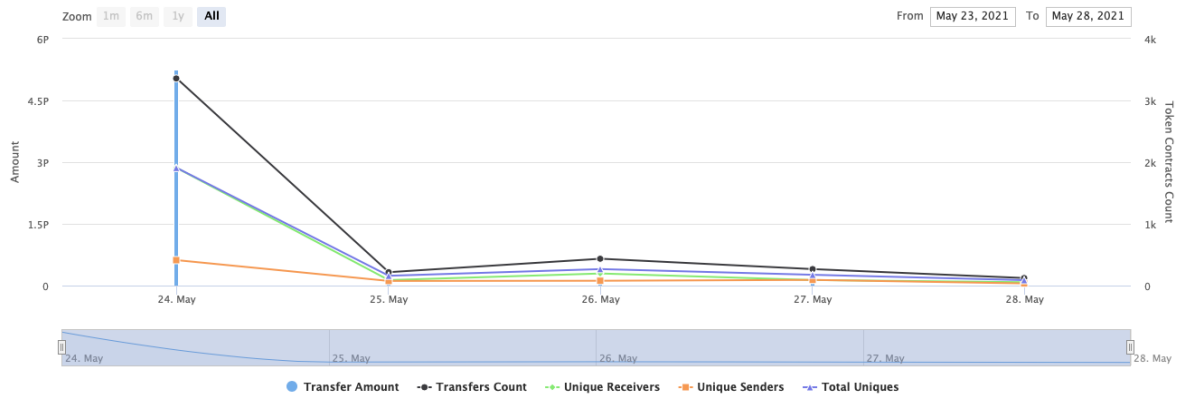
## BoggieCoin contract interaction details

Time Series: Token Contract Overview


Mon 24, May 2021 - Fri 28, May 2021

Token Contract 0x1570dD4DD36ad9E66D53179ecad33bF6dFD722b (BoggieCoin)

Source: BscScan.com



# BoggieCoin top 10 token holders

Rank	Address	Quantity (Token)	Percentage
1	0x0000000000000000000000000000000000dead	944,423,229,347,260.502529961	47.7972%
2	 0x47ef533877b3a8341c843d460cee4ed55455c017	233,896,164,417,477.953639791	11.8375%
3	0xb1b9239e6ce20a4e977f38e343d9ec2ff064ade1	106,014,638,173,439.372934409	5.3654%
4	0x32fd46d27e7843b8173f69f9418821ccd8d6a6d0	31,528,750,708,500.129216311	1.5957%
5	0xd78f3e0bf90c72fca6d336e2f108fdb9a3fb166	14,823,807,845,843.86121287	0.7502%
6	0xa7eeaa1b50ee18efe30dc1358c61a7930d7156e	14,010,919,645,061.92779316	0.7091%
7	0x93ea3d368d2c8646189f9b5a43694683ae76243d	13,800,044,601,115.123659984	0.6984%
8	0x009366b6d7efe54238cec4cc38068574f3a6cf7d	11,934,539,028,375.420206048	0.6040%
9	0x33f6ee932cea603fafd6854827259be172c91da4	9,500,045,543,191.35325753	0.4808%
10	0x7c2f67dbefe3b3b6c54e49dc0e3b62865641523b	7,550,881,995,092.57354342	0.3821%

# BoggieCoin LP token holders

Rank	Address	Quantity	Percentage
1	 0xeb3a9c56d963b971d320f889be2fb8b59853e449	3,488.846000255567	78.1848%
2	0x5b558f94540a4c5b1eff89d87ea26a8292bd31	961.492257976653132905	21.5470%
3	0x07d80ae6f36a5e08dca74ce884a24d39db9934ed	11.968200711098135687	0.2682%
4	 0x00	0.000000000000001	0.0000%

# Contract functions details

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

+ [Lib] Address

- [Int] isContract
- [Int] sendValue #
- [Int] functionCall #
- [Int] functionCall #
- [Int] functionCallWithValue #
- [Int] functionCallWithValue #
- [Priv] \_functionCallWithValue #

+ Ownable (Context)

- [Int] <Constructor> #
- [Pub] owner
- [Pub] renounceOwnership #
  - modifiers: onlyOwner
- [Pub] transferOwnership #
  - modifiers: onlyOwner
- [Pub] getUnlockTime
- [Pub] getTime
- [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] 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 #
- + BoggieCoin (Context, IERC20, Ownable)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Pub] isExcludedFromReward
  - [Pub] totalFees
  - [Pub] totalBurn
  - [Pub] minimumTokensBeforeSwapAmount
  - [Pub] deliver #
  - [Pub] reflectionFromToken
  - [Pub] tokenFromReflection
  - [Pub] excludeFromReward #
    - modifiers: onlyOwner
  - [Ext] includeInReward #
    - modifiers: onlyOwner
  - [Prv] \_approve #
  - [Prv] \_transfer #
  - [Prv] swapAndLiquify #
    - modifiers: lockTheSwap
  - [Prv] swapTokensForEth #
  - [Prv] addLiquidity #
  - [Prv] \_tokenTransfer #
  - [Prv] \_transferStandard #
  - [Prv] \_transferToExcluded #
  - [Prv] \_transferFromExcluded #
  - [Prv] \_transferBothExcluded #
  - [Prv] \_reflectFee #
  - [Prv] \_getValues
  - [Prv] \_getTValues
  - [Prv] \_getRValues

- [Prv] \_getRate
- [Prv] \_getCurrentSupply
- [Prv] \_takeLiquidity #
- [Prv] calculateTaxFee
- [Prv] calculateBurnFee
- [Prv] calculateLiquidityFee
- [Prv] removeAllFee #
- [Prv] restoreAllFee #
- [Pub] isExcludedFromFee
- [Pub] excludeFromFee #
  - modifiers: onlyOwner
- [Pub] includeInFee #
  - modifiers: onlyOwner
- [Ext] setTaxFeePercent #
  - modifiers: onlyOwner
- [Ext] setBurnFeePercent #
  - modifiers: onlyOwner
- [Ext] setLiquidityFeePercent #
  - modifiers: onlyOwner
- [Ext] setMaxTxPercent #
  - modifiers: onlyOwner
- [Ext] setNumTokensSellToAddToLiquidity #
  - modifiers: onlyOwner
- [Ext] setMarketingAddress #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Prv] transferForMarketingETH #
- [Ext] <Fallback> (\$)

(\$ ) = payable function

# = non-constant function

## Issues Checking Status

No	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(!_isExcluded[account], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            tOwned[account] = 0;
            _isExcluded[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() private view returns (uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            rOwned[_excluded[i]] > rSupply ||
            tOwned[_excluded[i]] > tSupply
        ) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(rOwned[_excluded[i]]);
        tSupply = tSupply.sub(tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
}
```

Recommendation:

Use `EnumerableSet` instead of array or do not use long arrays.

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

- ❑ Owner can change the tax, burn and liquidity fee.

```

function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}

function setBurnFeePercent(uint256 burnFee) external onlyOwner() {
    _burnFee = burnFee;
}

function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
    _liquidityFee = liquidityFee;
}

```

- ❑ Owner can change the maximum transaction amount.

```

function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner() {
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(
        10**2
    );
}

```

- ❑ Owner can exclude from the fee.

```

function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}

```

- ❑ Owner can change the marketing address to any address.

```

function setMarketingAddress(address payable _marketingAddress) external onlyOwner() {
    marketingAddress = _marketingAddress;
}

```

- ❑ Owner can lock and unlock. By the way, using these functions the owner could leave as owner even after the ownership was renounced.

```

//Locks the contract for owner for the amount of time provided
function lock(uint256 time) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = now + time;
    emit OwnershipTransferred(_owner, address(0));
}

//Unlocks the contract for owner when _lockTime is exceeds
function unlock() public virtual {
    require(_previousOwner == msg.sender, "You don't have permission to unlock");
    require(now > _lockTime, "Contract is locked until 7 days");
    emit OwnershipTransferred(_owner, _previousOwner);
    _owner = _previousOwner;
}

```

# Conclusion

Smart contracts contain low severity issues. LP pair contract is not checked.

Half of the liquidity adding transfers to the marketing address:

0xB1b9239E6Ce20a4E977F38e343d9ec2ff064Ade1

Liquidity locking details provided by the team:

<https://dxsale.app/app/pages/dxlockview?id=0&add=0x5b558F9454040A4C5b1Eff89d87ea26A8292bD31&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.*