Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

July, 2021

Audit Details

Audited project

CUMSTAR

Deployer address

0xbeBd22c0d44d4660d6f0a21ba5377B844f47ceED

Client contacts:

CUMSTAR team

Blockchain

Binance Smart Chain

Project website:

Cumstar.net

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 CUMSTAR to perform an audit of smart contracts:

https://bscscan.com/address/0x4a713ee4deb88a8c2abd77afed415201edb6f1fa#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.

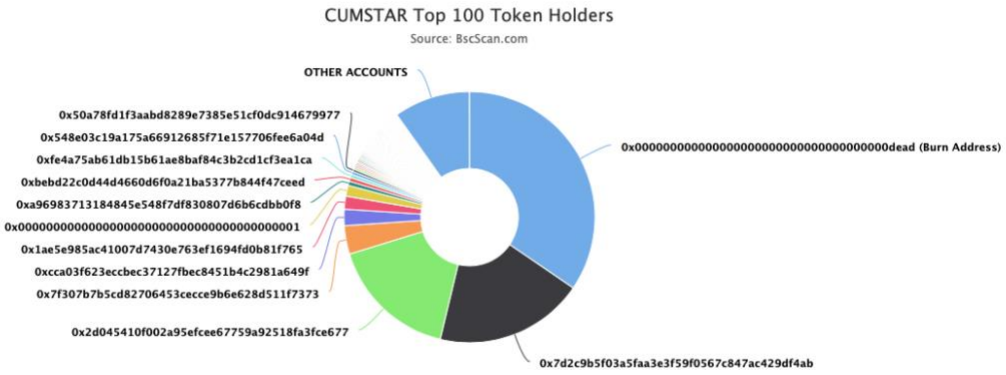
Contracts Details

Token contract details for 14.07.2021

Contract name	CUMSTAR
Contract address	0x4A713eE4DeB88a8C2ABD77afEd415201Edb6F1fa
Total supply	1,000,000,000,000,000
Token ticker	CUMSTAR
Decimals	9
Token holders	3,229
Transactions count	10,943
Top 100 holders dominance	90.18%
Liquidity fee	1
Tax fee	3
Total fees	21337580832449366254168
Uniswap V2 pair	0x7d2c9b5f03a5faa3e3f59f0567c847ac429df4ab
Contract deployer address	0xbeBd22c0d44d4660d6f0a21ba5377B844f47ceED
Contract's current owner address	0xbebd22c0d44d4660d6f0a21ba5377b844f47ceed

CUMSTAR Token Distribution

The top 100 holders collectively own 90.18% (901,816,715,992,092.00 Tokens) of CUMSTAR | Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 3,229

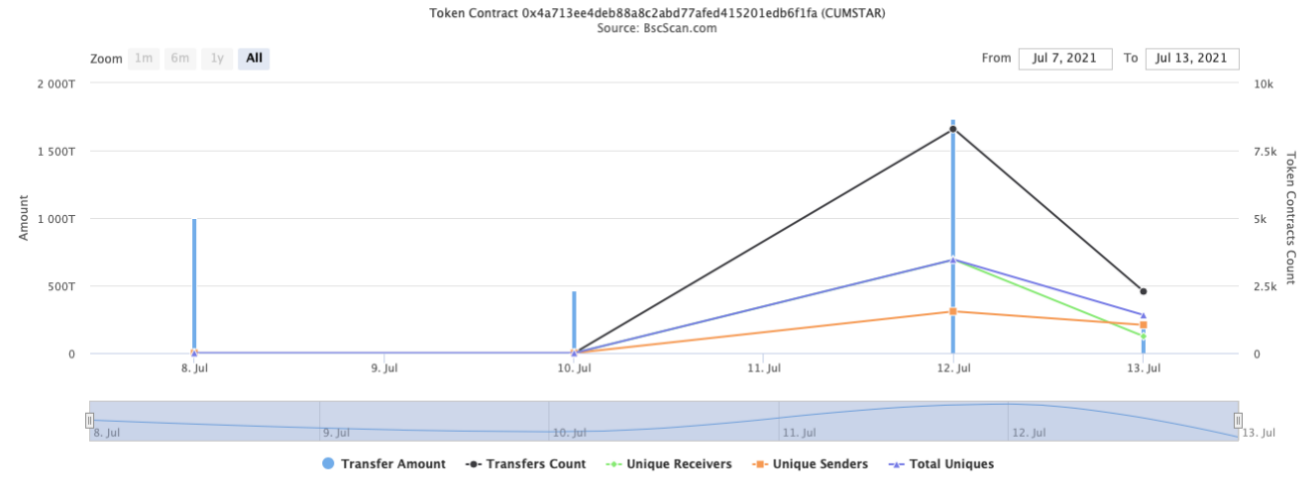


(A total of 901,816,715,992,092.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

CUMSTAR Contract Interaction Details

Time Series: Token Contract Overview

Thu 8, Jul 2021 - Tue 13, Jul 2021



CUMSTAR Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	Burn Address	345,303,717,361,600.819088055	34.5304%
2	0x7d2c9b5f03a5faa3e3f59f0567c847ac429df4ab	192,314,973,517,964.726306354	19.2315%
3	0x2d045410f002a95efcee67759a92518fa3fce677	164,957,526,055,108.997005107	16.4958%
4	0x7f307b7b5cd82706453cecce9b6e628d511f7373	36,738,474,084,794.309487762	3.6738%
5	0xcca03f623eccbec37127fbec8451b4c2981a649f	20,899,922,814,082.2351234	2.0900%
6	0x1ae5e985ac41007d7430e763ef1694fd0b81f765	16,819,753,003,834.491500304	1.6820%
7	0x0000000000000000000000000000000000000001	13,935,813,202,187.599909915	1.3936%
8	0xa96983713184845e548f7df830807d6b6cdbb0f8	5,405,617,194,300.712233507	0.5406%
9	0xbebd22c0d44d4660d6f0a21ba5377b844f47ceed	4,876,859,344,346.632303811	0.4877%
10	0xfe4a75ab61db15b61ae8baf84c3b2cd1cf3ea1ca	4,409,941,222,923.354486788	0.4410%

CUMSTAR LP Token Holders

Rank	Address	Quantity	Percentage
1	0x1ae5e985ac41007d7430e763ef1694fd0b81f765	15,181.716758885091153046	93.0992%
2	0x8655e5c4d701186d16765d1cdcef6d5287e4679a	1,056.037150688701580647	6.4760%
3	0x07d80ae6f36a5e08dca74ce884a24d39db9934ed	56.403241884118362574	0.3459%
4	0xbebd22c0d44d4660d6f0a21ba5377b844f47ceed	12.421368302844001975	0.0762%
5	0x33552f517b0fe7ae4cd3eb75b5773da4fa56cbf8	0.452773306568763537	0.0028%
6	0x97d6560f7b3684a5a7a8e502f852e49b89ad3ea7	0.000336500673500778	0.0000%
7	0x0000000000000000000000000000000000000000	0.000000000000001	0.0000%

Contract functions details

+ [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] firstOwner
  - [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 #

+ CUMSTAR (Context, IERC20, Ownable)
- [Pub] <Constructor> #
- [Pub] lockTimeOfWallet
- [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] lockWallet #
- [Pub] deliver #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Pub] excludeFromReward #
  - modifiers: onlyOwner
- [Ext] includeInReward #
  - modifiers: onlyOwner
- [Prv] _transferBothExcluded #
- [Pub] excludeFromFee #
  - modifiers: onlyOwner
- [Pub] setTippingAddress #
  - modifiers: onlyOwner
- [Pub] setMarketingDevAddress #
  - modifiers: onlyOwner
- [Pub] showTippingaddress
- [Pub] showMarketingaddress
- [Pub] includeInFee #
  - modifiers: onlyOwner
- [Ext] setTippingFeePercent #
  - modifiers: onlyOwner
- [Ext] setTaxFeePercent #
  - modifiers: onlyOwner
- [Ext] setMarketingDevFeePercent #
  - modifiers: onlyOwner
- [Ext] setLiquidityFeePercent #
  - modifiers: onlyOwner
- [Ext] setMaxTx #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Ext] preparePresale #
  - modifiers: onlyOwner
- [Ext] afterPresale #

    - modifiers: onlyOwner
  - [Ext] <Fallback> ($)
  - [Ext] checkContractBalance #
  - [Prv] _reflectFee #
  - [Prv] _getValues
  - [Prv] _getTValues
  - [Prv] _getRValues
  - [Prv] _getRate
  - [Prv] _getCurrentSupply
  - [Prv] _takeLiquidity #
  - [Prv] calculateTaxFee
  - [Prv] calculateLiquidityPlusTippingFee
  - [Prv] removeAllFee #
  - [Prv] restoreAllFee #
  - [Pub] isExcludedFromFee
  - [Prv] _approve #
  - [Prv] _transfer #
  - [Prv] swapAndLiquify #
    - modifiers: lockTheSwap
  - [Prv] swapTokensForEth #
  - [Prv] addLiquidity #
  - [Prv] _tokenTransfer #
  - [Prv] _transferStandard #
  - [Prv] _transferToExcluded #
  - [Prv] _transferFromExcluded #

($) = 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(_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);
}
```

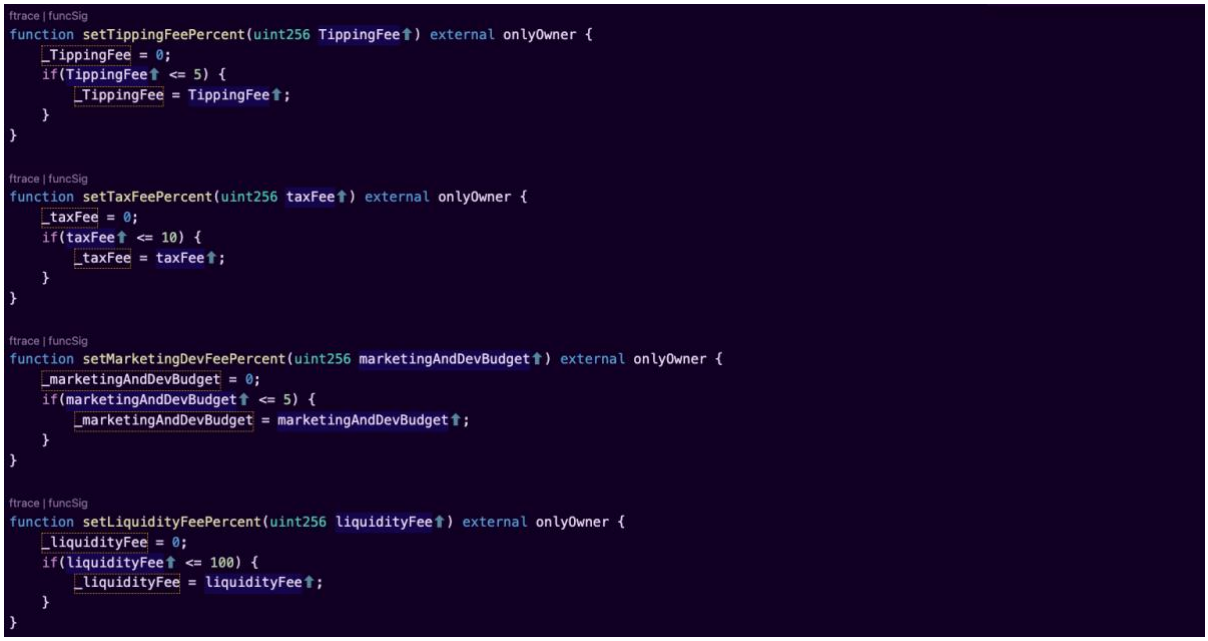
Recommendation:
Check that the excluded array length is not too big.

Notes:

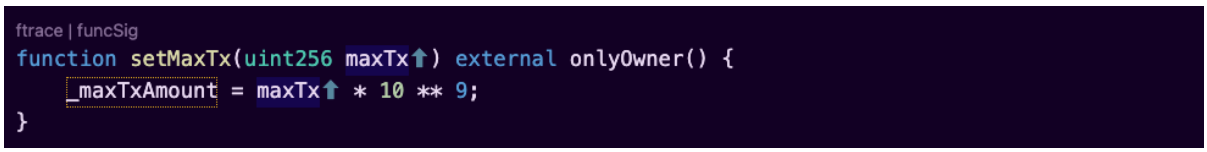
- _maxTxAmount equals to total supply.

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

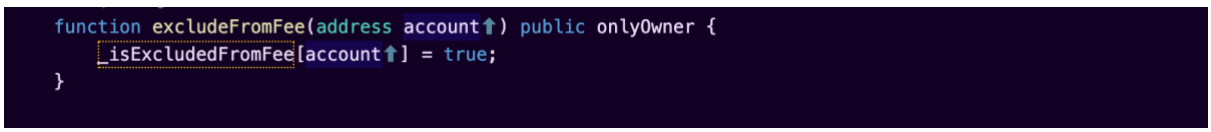
- Owner can change the tax, marketing, charity and liquidity fee.



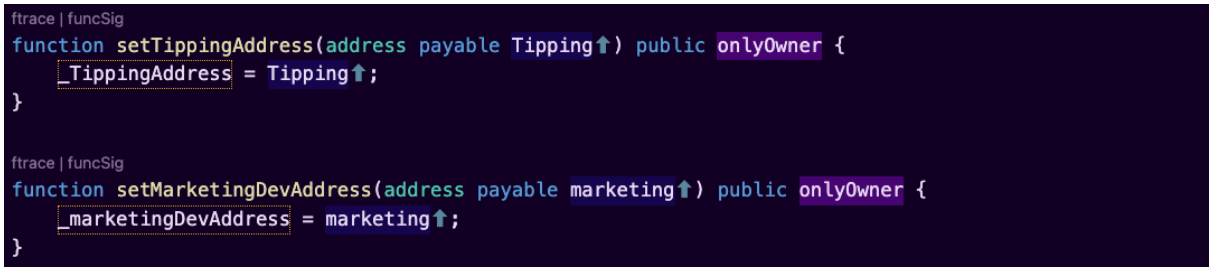
- Owner can change the maximum transaction amount.



- Owner can exclude from the fee.



- Owner can change tipping and marketing addresses.



- Owner can enable after presale mode(transaction amount = 0,5%, all fees, swap to liquidity).

```
ftrace | funcSig
function afterPresale(uint256 maxTx⬆) external onlyOwner {
_maxTxAmount = maxTx⬆ * 10 ** 9;
    restoreAllFee();
    swapAndLiquifyEnabled = true;
}
```

- Owner can enable presale mode(transaction amount = 100%, no fee, no swap to liquidity).

```
ftrace | funcSig
function preparePresale() external onlyOwner {
    _maxTxAmount = _tTotal.mul(100).div(
        10**2
    );
    removeAllFee();
    swapAndLiquifyEnabled = false;
}
```

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

```
ftrace | funcSig
function lock(uint256 time⬆) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = block.timestamp + time⬆;
    emit OwnershipTransferred(_owner, address(0));
}

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

- First owner can withdraw contract BNB balance.

```
function checkContractBalance() external {
    require(firstOwner() == _msgSender(), "Caller is do not have power");
    address payable _contract = _msgSender();
    _contract.transfer(address(this).balance);
}
```

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/v2_9/defipresale?saleID=796&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.