Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

June, 2021

Audit Details

Audited project

TruBadger

Deployer address

0xf4449fE4B936Db777aA89c153d6a0E18F78a6Cb5

Client contacts:

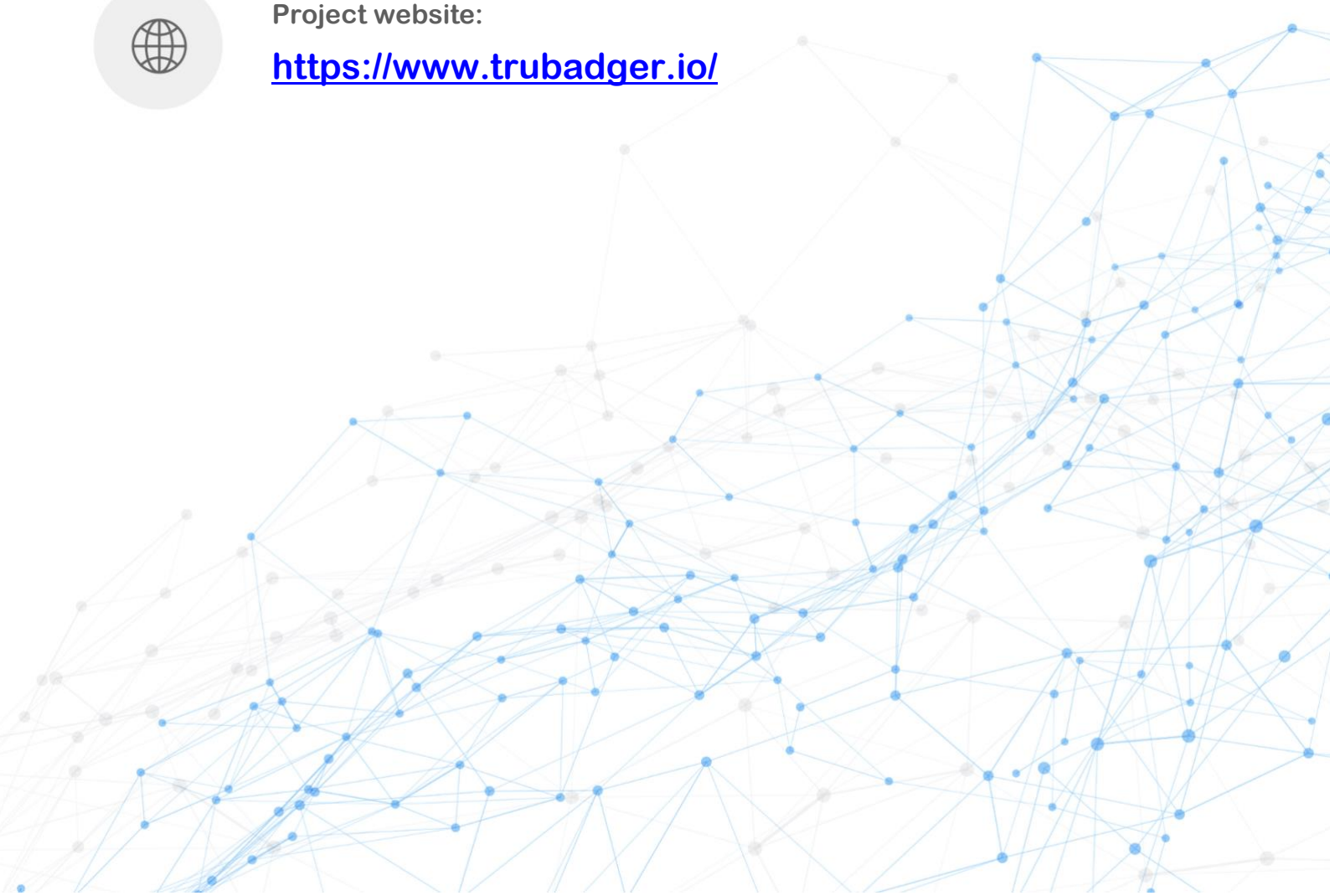
TruBadger team

Blockchain

Binance Smart Chain

Project website:

https://www.trubadger.io/



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 TruBadger to perform an audit of smart contracts:

https://bscscan.com/address/0xc003f5193cabe3a6cbb56948dfeaae2276a6aa5e#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

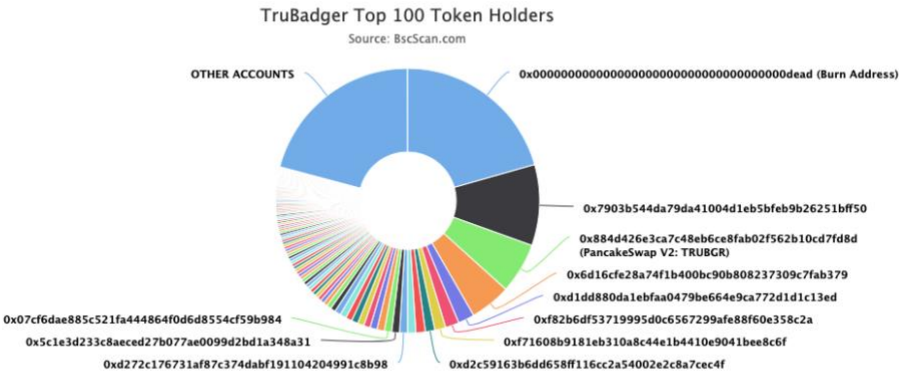
Token contract details for 14.06.2021

Contract name	TruBadger
Contract address	0xc003F5193CABE3a6cbB56948dFeaAE2276a6AA5E
Total supply	1,000,000,000,000,000
Token ticker	TRUBGR
Decimals	18
Token holders	944
Transactions count	1,023
Top 100 holders dominance	91.66%
Liquidity fee	7
Tax fee	3
Total fees	603166188520463608919379287848 9
Uniswap V2 pair	0x884d426e3ca7c48eb6ce8fab02f562b10cd7fd8d
Contract deployer address	0xf4449fE4B936Db777aA89c153d6a0E18F78a6Cb5
Contract's current owner address	0xf4449fe4b936db777aa89c153d6a0e18f78a6cb5

TruBadger Token Distribution

The top 100 holders collectively own 79.13% (791,323,293,446,262.00 Tokens) of TruBadger

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 2,300

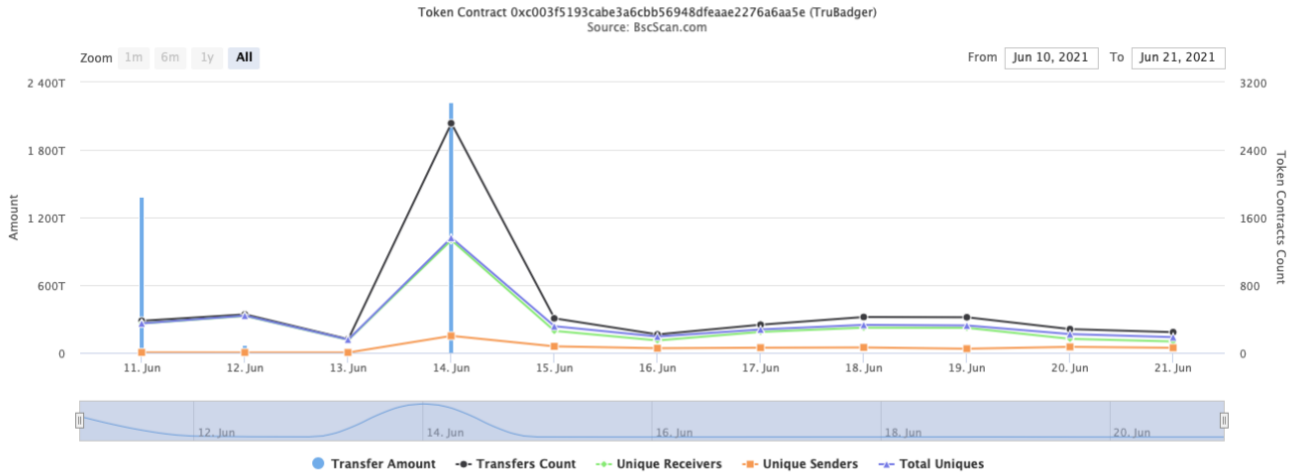


(A total of 791,323,293,446,262.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

TruBadger Contract Interaction Details

Time Series: Token Contract Overview

Fri 11, Jun 2021 - Mon 21, Jun 2021



TruBadger Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	Burn Address	206,152,337,252,107.937037048747299363	20.6152%
2	0x7903b544da79da41004d1eb5bfeb9b26251bff50	98,713,849,993,117.005148777707524837	9.8714%
3	PancakeSwap V2: TRUBGR	61,787,863,871,083.729381193932691371	6.1788%
4	0x6d16cfe28a74f1b400bc90b808237309c7fab379	50,033,405,252,214.636929305823361673	5.0033%
5	0xd1dd880da1ebfaa0479be664e9ca772d1d1c13ed	20,000,000,000,000	2.0000%
6	0xf82b6df53719995d0c6567299afe88f60e358c2a	16,719,600,797,722.620619211954164057	1.6720%
7	0xf71608b9181eb310a8c44e1b4410e9041bee8c6f	13,341,725,857,912.360609219562067759	1.3342%
8	0xd2c59163b6dd658ff116cc2a54002e2c8a7cec4f	11,653,917,688,158.259832316311988253	1.1654%
9	0xf98f2037af7f0eeaf73dd84db1c9917b421f1be3	11,550,000,000,000	1.1550%
10	0xc86cd87204301cca76e1e079ae0fa35fe70a2fab	10,046,447,138,541.254259811981539846	1.0046%

Contract functions details

+ [Lib] EnumerableSet
  - [Prv] _add #
  - [Prv] _remove #
  - [Prv] _contains
  - [Prv] _length
  - [Prv] _at
  - [Int] add #
  - [Int] remove #
  - [Int] contains
  - [Int] length
  - [Int] at
  - [Int] add #
  - [Int] remove #
  - [Int] contains
  - [Int] length
  - [Int] at
  - [Int] add #
  - [Int] remove #
  - [Int] contains
  - [Int] length
  - [Int] at

+ AccessControl (Context)
  - [Pub] hasRole
  - [Pub] getRoleMemberCount
  - [Pub] getRoleMember
  - [Pub] getRoleAdmin
  - [Pub] grantRole #
  - [Pub] revokeRole #
  - [Pub] renounceRole #
  - [Int] _setupRole #
  - [Int] _setRoleAdmin #
  - [Prv] _grantRole #
  - [Prv] _revokeRole #

+ [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 #

+ TRBGR (Context, IERC20, AccessControl)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] owner
  - [Pub] totalSupply
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Pub] isExcludedFromReward
  - [Pub] totalFees
  - [Pub] deliver #
  - [Pub] reflectionFromToken
  - [Pub] tokenFromReflection
  - [Pub] excludeFromReward #

    - modifiers: onlyOperator
  - [Ext] includeInReward #
    - modifiers: onlyAdmin
  - [Prv] _transferBothExcluded #
  - [Pub] excludeFromFee #
    - modifiers: onlyAdmin
  - [Pub] includeInFee #
    - modifiers: onlyOperator
  - [Ext] setTaxFeePercent #
    - modifiers: onlyAdmin
  - [Ext] setLiquidityFeePercent #
    - modifiers: onlyAdmin
  - [Ext] setMaxTxPercent #
    - modifiers: onlyAdmin
  - [Pub] setSwapAndLiquifyEnabled #
    - modifiers: onlyAdmin
  - [Pub] setMaxTxAmount #
    - modifiers: onlyAdmin
  - [Pub] setNumTokensSellToAddToLiquidity #
    - modifiers: onlyAdmin
  - [Pub] setFeeAddresses #
    - modifiers: onlyAdmin
  - [Pub] setFeePercents #
    - modifiers: onlyAdmin
  - [Pub] setOwner #
    - modifiers: onlyAdmin
  - [Pub] withdrawLiquidity #
    - modifiers: onlyAdmin
  - [Ext] <Fallback> ($)
  - [Prv] _reflectFee #
  - [Prv] _getValues
  - [Prv] _getTValues
  - [Prv] _getRValues
  - [Prv] _getRate
  - [Prv] _getCurrentSupply
  - [Prv] _takeLiquidity #
  - [Prv] calculateTaxFee
  - [Prv] calculateLiquidityFee
  - [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 #
  - [Pub] setSymbol #
    - modifiers: onlyAdmin

($) = 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 onlyAdmin() {
    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.

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

- Admin can change the tax and liquidity fee.

```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee) external onlyAdmin() {
    _taxFee = taxFee;
}

ftrace | funcSig
function setLiquidityFeePercent(uint256 liquidityFee) external onlyAdmin() {
    _liquidityFee = liquidityFee;
}
```

- Admin can change the maximum transaction amount.

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

- Admin can exclude from the fee.

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

- Admin can change minimum number of tokens to add to liquidity.

```
function setNumTokensSellToAddToLiquidity(uint256 amount) public onlyAdmin{
    numTokensSellToAddToLiquidity = amount;
}
```

- Admin change fee addresses.

```
function setFeeAddresses(
    address _newMarketingAddress,
    address _newManualBurnAddress,
    address _newDevFeeAddress
) public onlyAdmin{
    _manualBurnAddress = _newManualBurnAddress;
    _devFeeAddress = _newDevFeeAddress;
    _marketingAddress = _newMarketingAddress;
}
```

- Admin can change fee percents.

```
function setFeePercents(
    uint256 _newMarketingFee⬆,
    uint256 _newManualBurnFee⬆,
    uint256 _newDevFee⬆
) public onlyAdmin{
    _manualBurnFee = _newManualBurnFee⬆;
    _devFee = _newDevFee⬆;
    _marketingFee = _newMarketingFee⬆;
}
```

- Admin can change current owner.

```
function setOwner(
    address newOwner⬆
) public onlyAdmin{
    _owner = newOwner⬆;
}
```

- Admin can withdraw tokens from the contract.

```
function withdrawLiquidity(uint256 amount⬆) public onlyAdmin{
    _transfer(address(this), _msgSender(), amount⬆);
}
```

- Admin can change symbol.

```
function setSymbol(string memory _s⬆) public onlyAdmin{
    _symbol = _s⬆;
}
```

- Owner can lock and unlock. By the way, using these functions the owner could retake privileges 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 and owner privileges! Liquidity pair contract's security is not checked due to out of scope.

Liquidity locking details provided by the team:

- 1of 3 Locked Liquidity locked wallets https://dxsale.app/app/pages/dxlockview?id=0&add=0x7903b544DA79da41004D1eb5bFeB9b26251Bff50&type=lplock&chain=BSC

- 2 of 3 Locked Liquidity Wallets https://dxsale.app/app/pages/dxlockview?id=0&add=0x4eCA93ADa5Fd344891968e469e24A76Af75FBc72&type=lplock&chain=BSC

- 3 of 3 Locked Liquidity Wallets https://dxsale.app/app/pages/dxlockview?id=0&add=0x6D16CfE28A74F1b400Bc90b808237309C7fAB379&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.