Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

June, 2021

Audit Details

Audited project

Altrucoin

Deployer address

0x25A7aB5a0A175688adE82bC452A638B0E964EdAD

Client contacts:

Altrucoin team

Blockchain

Binance Smart Chain

Project website:

Not provided by Altrucoin team

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 Altrucoin to perform an audit of smart contracts:

https://bscscan.com/address/0xeDAF1F5B8078d4feb4E13c8d5A2c8dE1365be7b6#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.

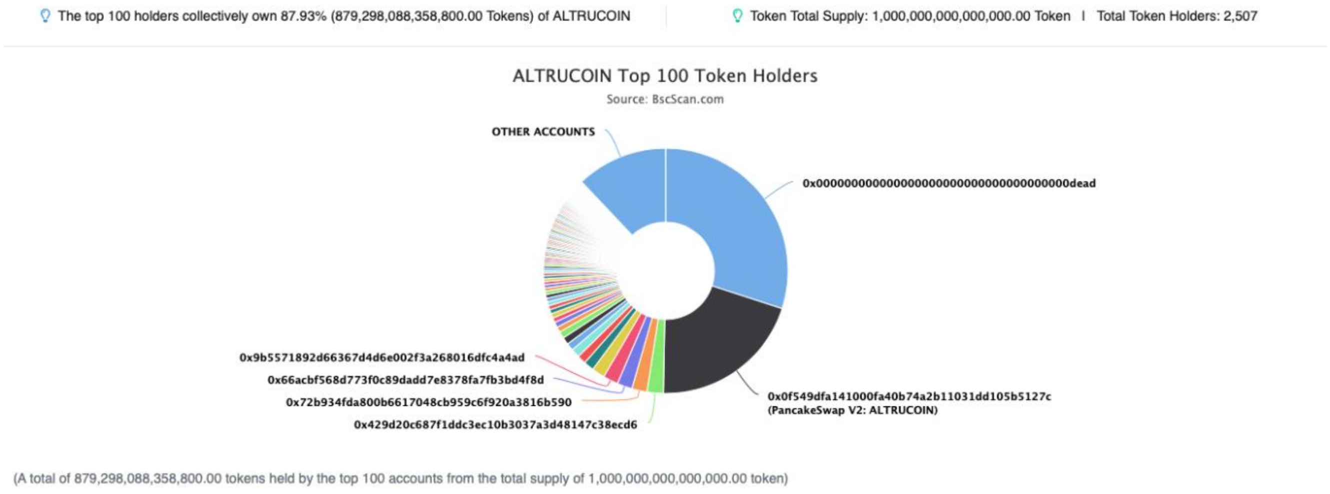
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 08.06.2021

Contract name	Altrucoin
Contract address	0xeDAF1F5B8078d4feb4E13c8d5A2c8dE1365be7b6
Total supply	1,000,000,000,000,000
Token ticker	ALTRUCOIN
Decimals	9
Token holders	2,507
Transactions count	10,856
Top 100 holders dominance	87.93%
Liquidity fee	2
Tax fee	5
Total fees	110333184194501855991035
Uniswap V2 pair	0x0f549dfa141000fa40b74a2b11031dd105b5127c
Contract deployer address	0x25A7aB5a0A175688adE82bC452A638B0E964EdAD
Contract's current owner address	0xcf6b15bd93afacc413352aa5b20bd7724d253d47

Altrucoin Token Distribution



The top 100 holders collectively own 87.93% (879,298,088,358,800.00 Tokens) of ALTRUCOIN

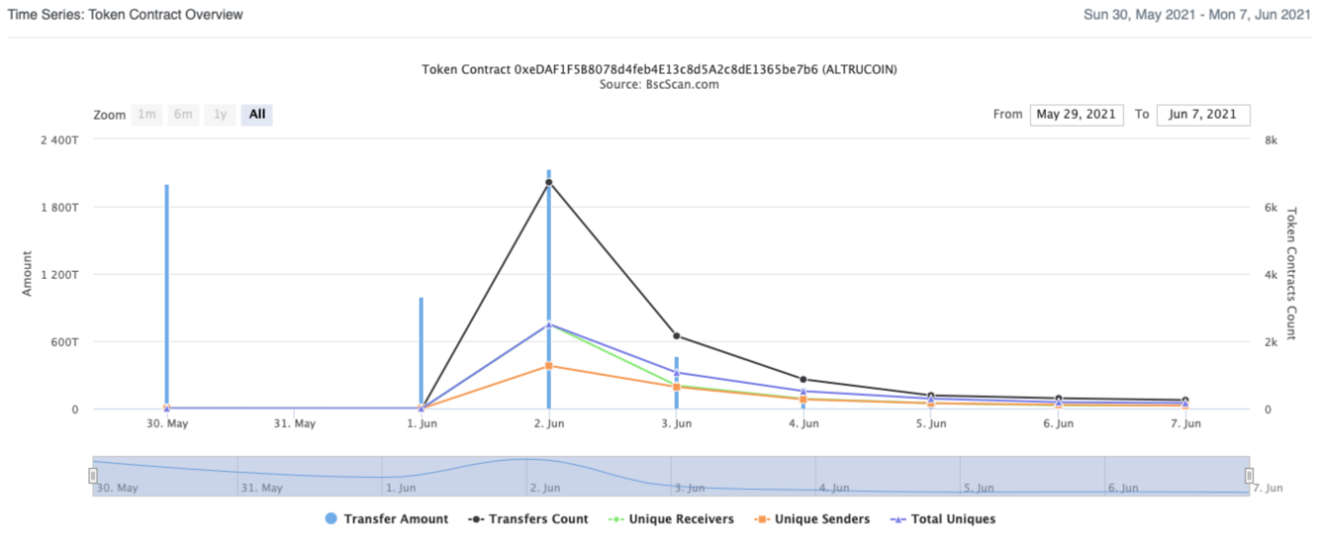
Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 2,507

ALTRUCOIN Top 100 Token Holders

Source: BscScan.com

(A total of 879,298,088,358,800.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

Altrucoin Contract Interaction Details



Time Series: Token Contract Overview

Sun 30, May 2021 - Mon 7, Jun 2021

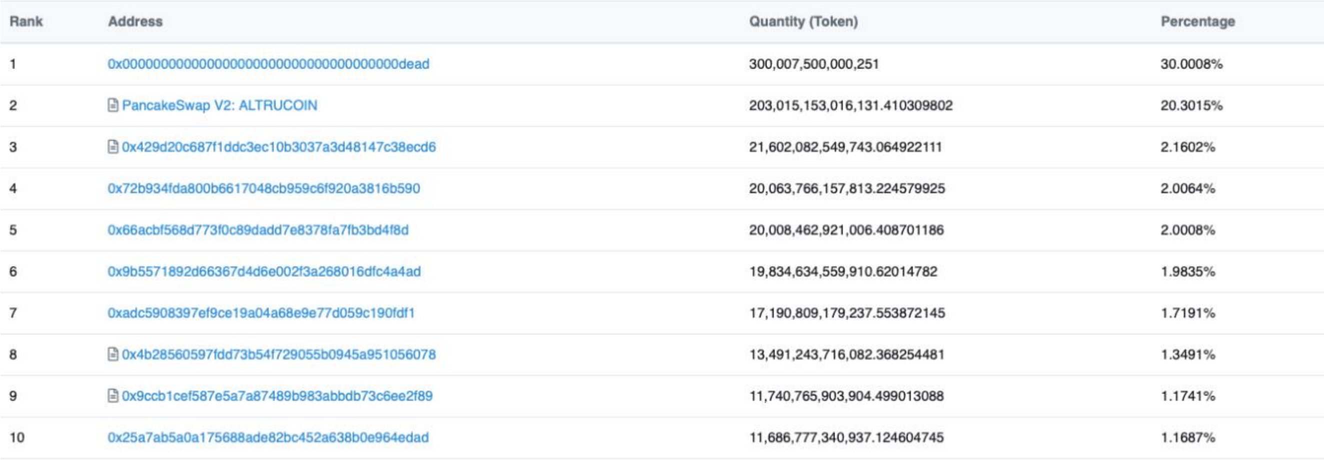
Token Contract 0xeDAF1F5B8078d4feb4E13c8d5A2c8dE1365be7b6 (ALTRUCOIN)

Source: BscScan.com

Zoom 1m 6m 1y All

From May 29, 2021 To Jun 7, 2021

Altrucoin Top 10 Token Holders



Rank	Address	Quantity (Token)	Percentage
1	0x000000000000000000000000000000000000dead	300,007,500,000,251	30.0008%
2	📄 PancakeSwap V2: ALTRUCOIN	203,015,153,016,131.410309802	20.3015%
3	📄 0x429d20c687f1ddc3ec10b3037a3d48147c38ecd6	21,602,082,549,743.064922111	2.1602%
4	0x72b934fda800b6617048cb959c6f920a3816b590	20,063,766,157,813.224579925	2.0064%
5	0x66acbf568d773f0c89dadd7e8378fa7fb3bd4f8d	20,008,462,921,006.408701186	2.0008%
6	0x9b5571892d66367d4d6e002f3a268016dfc4a4ad	19,834,634,559,910.62014782	1.9835%
7	0xadc5908397ef9ce19a04a68e9e77d059c190fdf1	17,190,809,179,237.553872145	1.7191%
8	📄 0x4b28560597fdd73b54f729055b0945a951056078	13,491,243,716,082.368254481	1.3491%
9	📄 0x9ccb1cef587e5a7a87489b983abbdb73c6ee2f89	11,740,765,903,904.499013088	1.1741%
10	0x25a7ab5a0a175688ade82bc452a638b0e964edad	11,686,777,340,937.124604745	1.1687%

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] 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 #

+ Altrucoin (Context, IERC20, Ownable)
  - [Pub] <Constructor> #
  - [Pub] setRouterAddressAndCreatePair #
    - modifiers: onlyOwner
  - [Pub] setRouterAddress #
    - modifiers: onlyOwner
  - [Pub] setPairAddress #
    - modifiers: onlyOwner
  - [Pub] setCharityAddress #
    - modifiers: onlyOwner
  - [Pub] setDevelopmentAddress #
    - modifiers: onlyOwner
  - [Pub] setMarketingAddress #
    - modifiers: onlyOwner
  - [Pub] setLiquidityAddress #
    - modifiers: onlyOwner
  - [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] deliver #
  - [Pub] reflectionFromToken
  - [Pub] tokenFromReflection
  - [Pub] excludeFromReward #
    - modifiers: onlyOwner
  - [Ext] includeInReward #
    - modifiers: onlyOwner
  - [Prv] _transferBothExcluded #
  - [Pub] excludeFromFee #
    - modifiers: onlyOwner
  - [Pub] includeInFee #
    - modifiers: onlyOwner
  - [Ext] setTaxFeePercent #
    - modifiers: onlyOwner
  - [Ext] setCharityFeePercent #
    - modifiers: onlyOwner
  - [Ext] setDevelopmentFeePercent #
    - modifiers: onlyOwner
  - [Ext] setMarketingFeePercent #
    - modifiers: onlyOwner
  - [Ext] setLiquidityFeePercent #
    - modifiers: onlyOwner
```

- [Ext] setMaxTxPercent #
  - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
  - modifiers: onlyOwner
- [Ext] <Fallback> ($)
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _takeLiquidity #
- [Prv] _takeCharity #
- [Prv] _takeMarketing #
- [Prv] _takeDevelopmentFee #
- [Prv] calculateTaxFee
- [Prv] calculateCharityFee
- [Prv] calculateMarketingFee
- [Prv] calculateDevelopmentFee
- [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 #

($) = 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

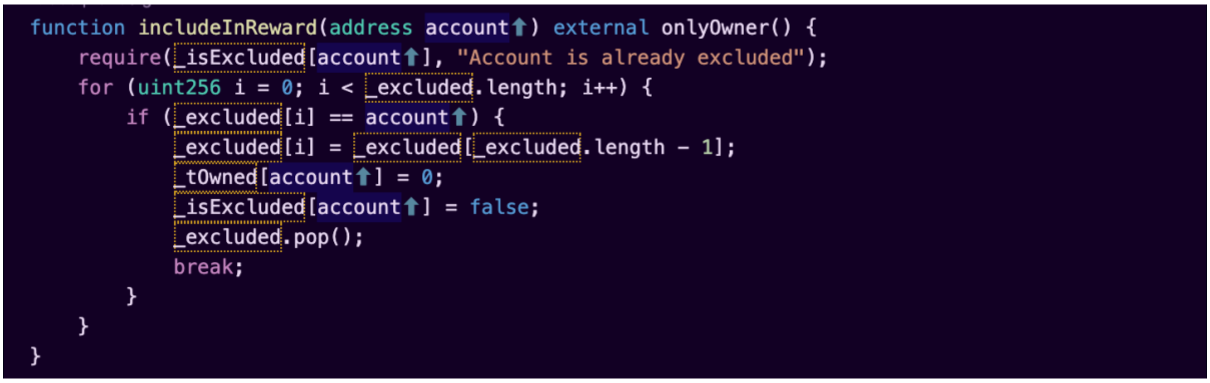
No medium severity issues found.

Low Severity Issues

1. Out of gas

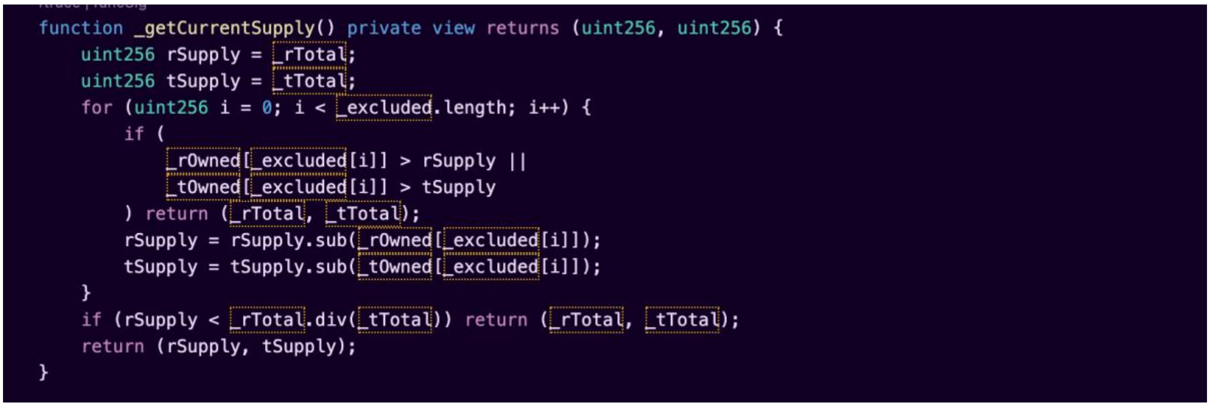
Issue:

- 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.



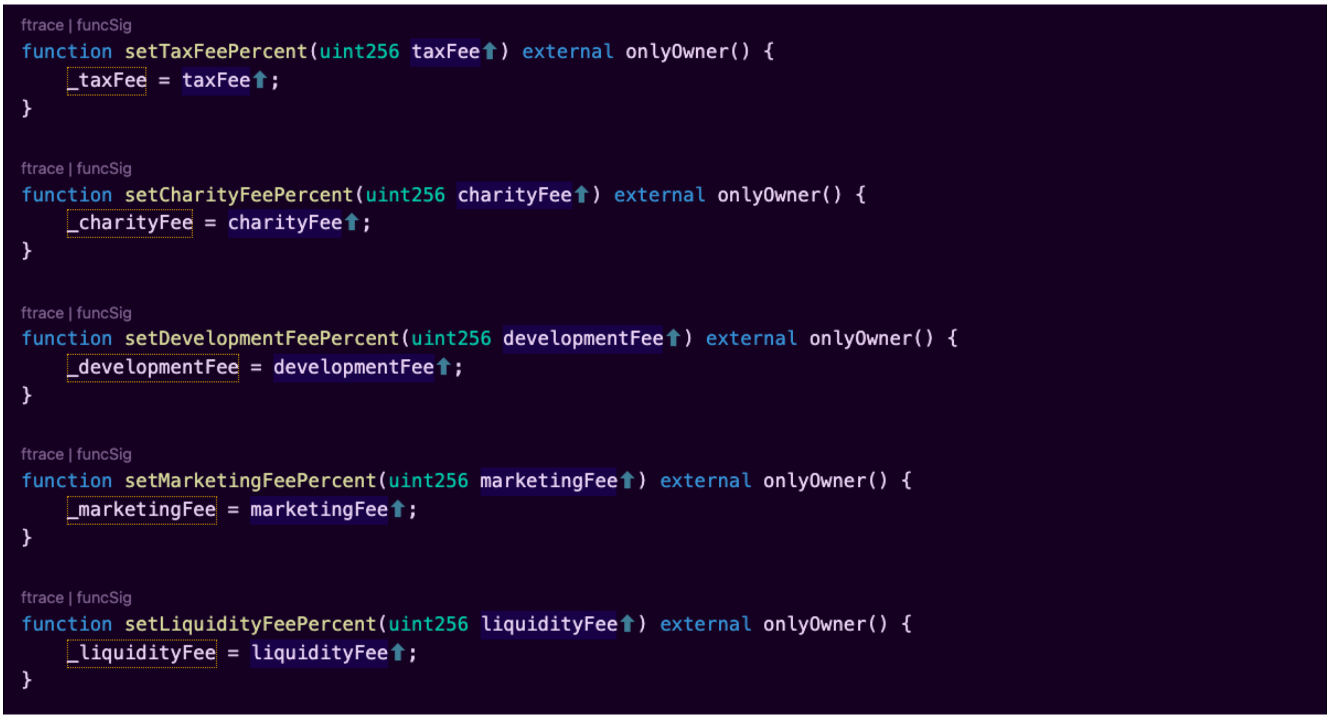
```
function _getCurrentSupply() private view returns (uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _rOwned[_excluded[i]] > rSupply ||
            _tOwned[_excluded[i]] > tSupply
        ) 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.

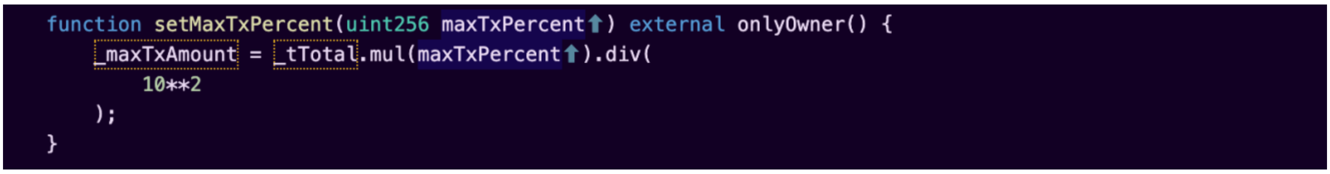
Owner privileges (In the period when the owner is not renounced)

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



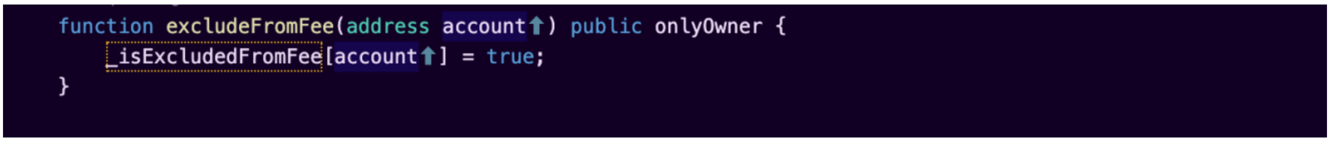
```
ftrace | funcSig
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}

ftrace | funcSig
function setCharityFeePercent(uint256 charityFee) external onlyOwner() {
    _charityFee = charityFee;
}

ftrace | funcSig
function setDevelopmentFeePercent(uint256 developmentFee) external onlyOwner() {
    _developmentFee = developmentFee;
}

ftrace | funcSig
function setMarketingFeePercent(uint256 marketingFee) external onlyOwner() {
    _marketingFee = marketingFee;
}

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

- Owner can change the maximum transaction amount.



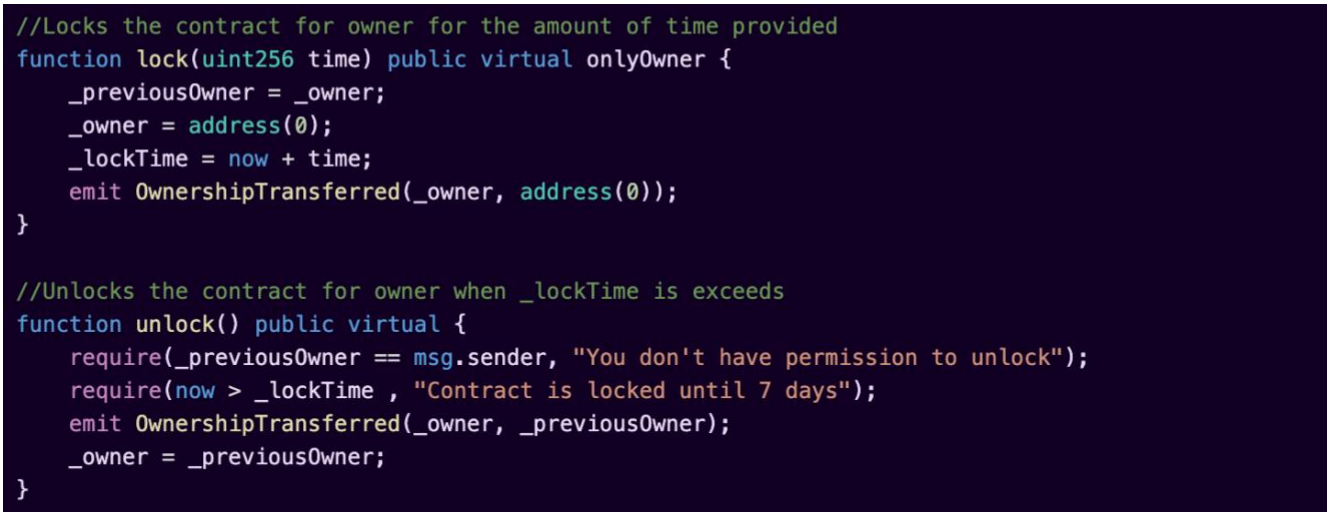
```
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner() {
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(
        10**2
    );
}
```

- Owner can exclude from the fee.



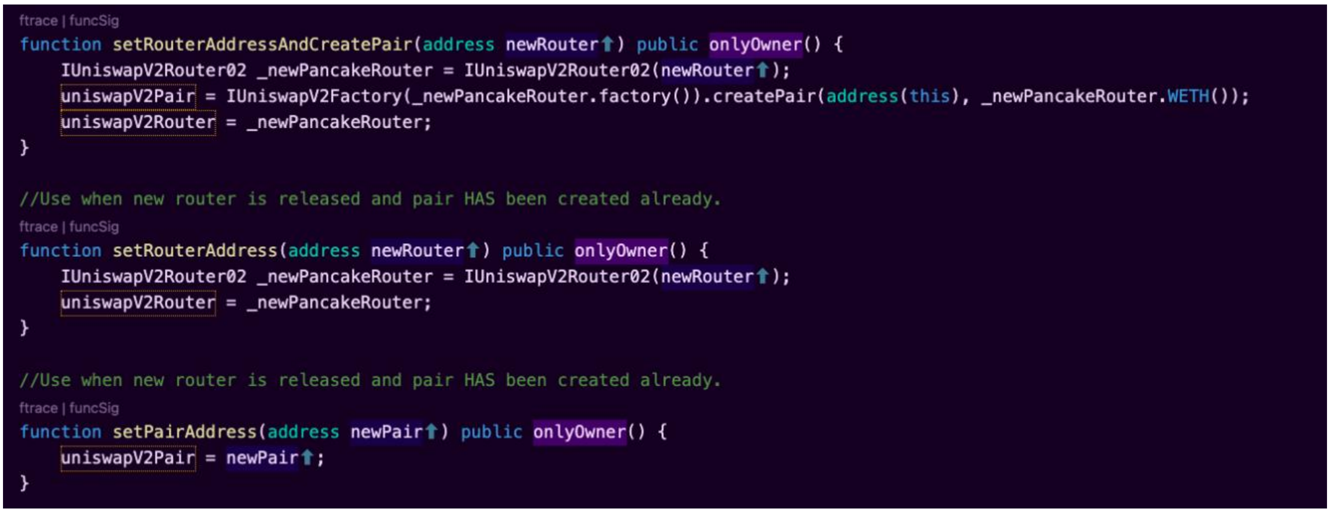
```
function excludeFromFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = true;
}
```

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



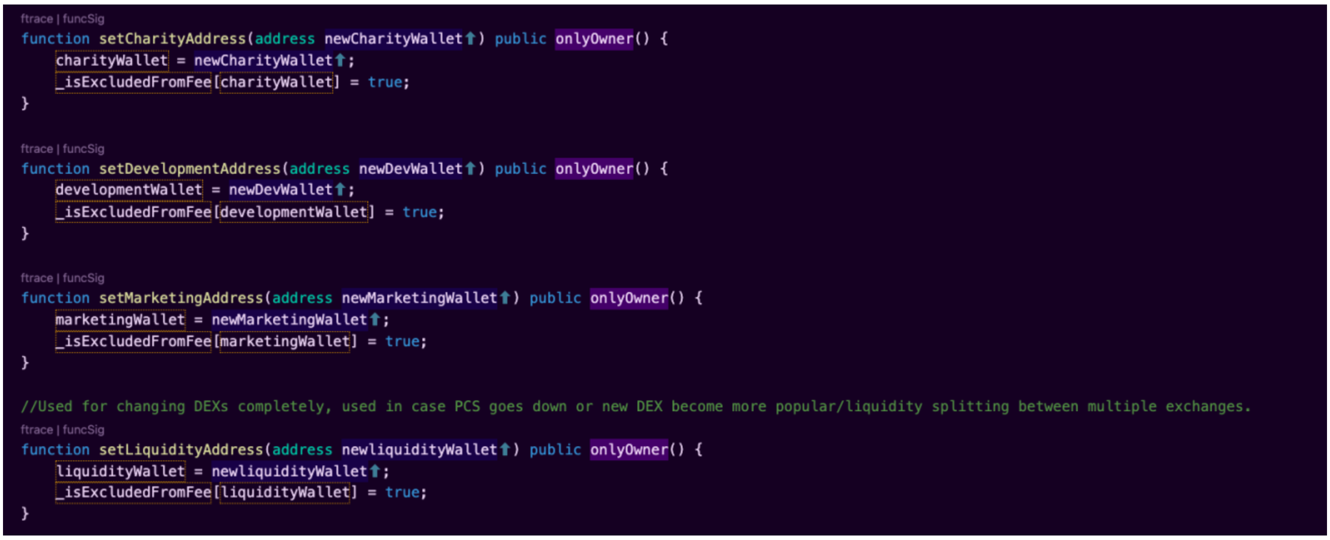
```
//Locks the contract for owner for the amount of time provided
function lock(uint256 time) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = now + time;
    emit OwnershipTransferred(_owner, address(0));
}

//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;
}
```

- **Owner can change Uniswap router and pair.**



```
ftrace | funcSig
function setRouterAddressAndCreatePair(address newRouter) public onlyOwner() {
    IUniswapV2Router02 _newPancakeRouter = IUniswapV2Router02(newRouter);
    uniswapV2Pair = IUniswapV2Factory(_newPancakeRouter.factory()).createPair(address(this), _newPancakeRouter.WETH());
    uniswapV2Router = _newPancakeRouter;
}

//Use when new router is released and pair HAS been created already.
ftrace | funcSig
function setRouterAddress(address newRouter) public onlyOwner() {
    IUniswapV2Router02 _newPancakeRouter = IUniswapV2Router02(newRouter);
    uniswapV2Router = _newPancakeRouter;
}

//Use when new router is released and pair HAS been created already.
ftrace | funcSig
function setPairAddress(address newPair) public onlyOwner() {
    uniswapV2Pair = newPair;
}
```

- **Owner can change charity, development, marketing and liquidity addresses.**



```
ftrace | funcSig
function setCharityAddress(address newCharityWallet) public onlyOwner() {
    charityWallet = newCharityWallet;
    _isExcludedFromFee[charityWallet] = true;
}

ftrace | funcSig
function setDevelopmentAddress(address newDevWallet) public onlyOwner() {
    developmentWallet = newDevWallet;
    _isExcludedFromFee[developmentWallet] = true;
}

ftrace | funcSig
function setMarketingAddress(address newMarketingWallet) public onlyOwner() {
    marketingWallet = newMarketingWallet;
    _isExcludedFromFee[marketingWallet] = true;
}

//Used for changing DEXs completely, used in case PCS goes down or new DEX become more popular/liquidity splitting between multiple exchanges.
ftrace | funcSig
function setLiquidityAddress(address newliquidityWallet) public onlyOwner() {
    liquidityWallet = newliquidityWallet;
    _isExcludedFromFee[liquidityWallet] = true;
}
```

Conclusion

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

Liquidity locking details NOT provided by the team.

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.