Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details

Audited project

ROBO INU

Deployer address

0xa116abd1b09c0b32b60260e2a65a3063aff78b75

Client contacts:

ROBO INU team

Blockchain

Ethereum

Project website:

https://roboinu.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 ROBO INU to perform an audit of smart contracts:

https://etherscan.io/address/0x7b32e70e8d73ac87c1b342e063528b2930b15ceb#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.

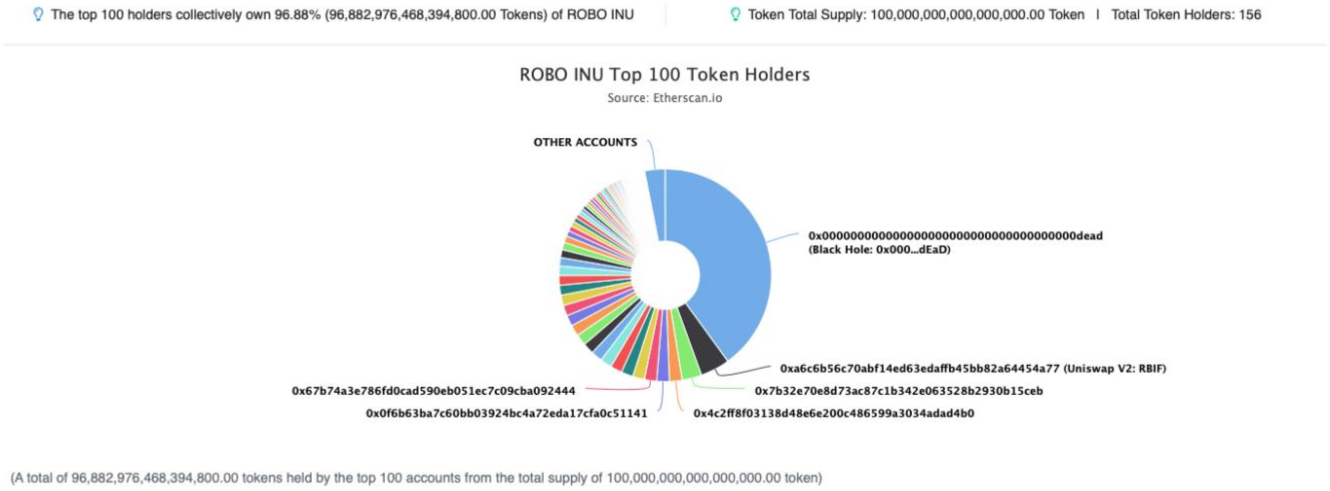
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 15.11.2021

Contract name	Token
Contract address	0x7B32e70e8D73ac87c1B342e063528B2930b15ceB
Total supply	100,000,000,000,000,000
Token ticker	RBIF
Decimals	9
Token holders	156
Transactions count	708
Top 100 holders dominance	96.88%
Liquidity fee	2
Tax fee	1
Total fees	80741215948836507276579 51
Uniswap V2 pair	0xa6c6b56c70abf14ed63edaffb45bb82a64454a77
Contract deployer address	0xa116abd1b09c0b32b60260e2a65a3063aff78b75
Contract’s current owner address	0xa116abd1b09c0b32b60260e2a65a3063aff78b75

ROBO INU Token Distribution



The top 100 holders collectively own 96.88% (96,882,976,468,394,800.00 Tokens) of ROBO INU

Token Total Supply: 100,000,000,000,000,000.00 Token | Total Token Holders: 156

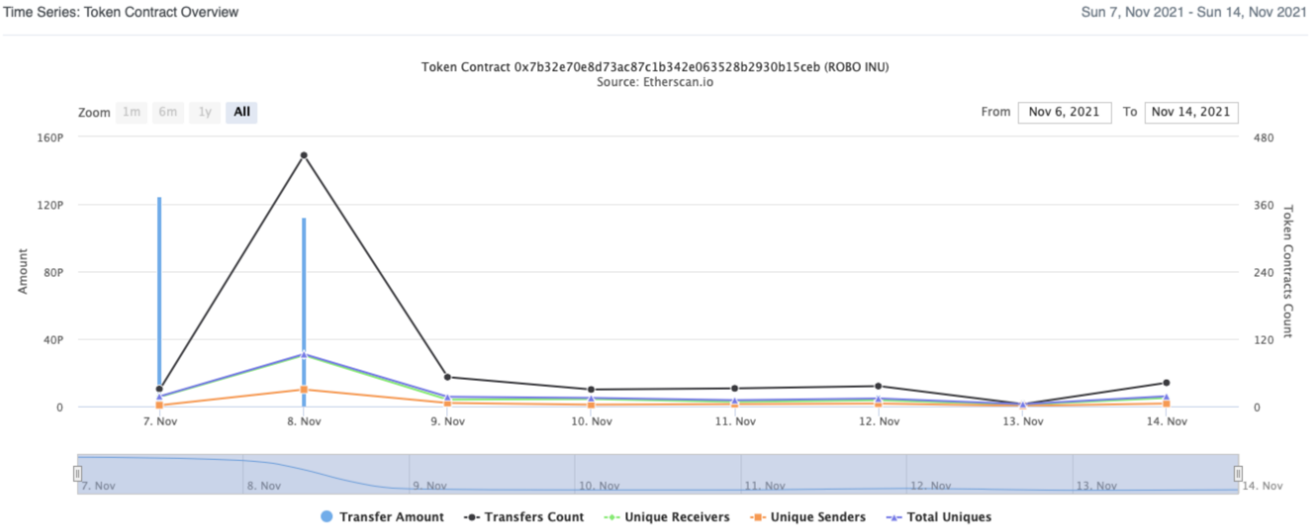
ROBO INU Top 100 Token Holders

Source: Etherscan.io

OTHER ACCOUNTS
0x000000000000000000000000000000000000dead
(Black Hole: 0x000...dEaD)
0xa6c6b56c70abf14ed63edaffb45bb82a64454a77 (Uniswap V2: RBIF)
0x67b74a3e786fd0cad590eb051ec7c09cba092444
0x7b32e70e8d73ac87c1b342e063528b2930b15ceb
0x0f6b63ba7c60bb03924bc4a72eda17cfa0c51141
0x4c2ff8f03138d48e6e200c486599a3034adad4b0

(A total of 96,882,976,468,394,800.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000,000.00 token)

ROBO INU Contract Interaction Details



Time Series: Token Contract Overview

Sun 7, Nov 2021 - Sun 14, Nov 2021

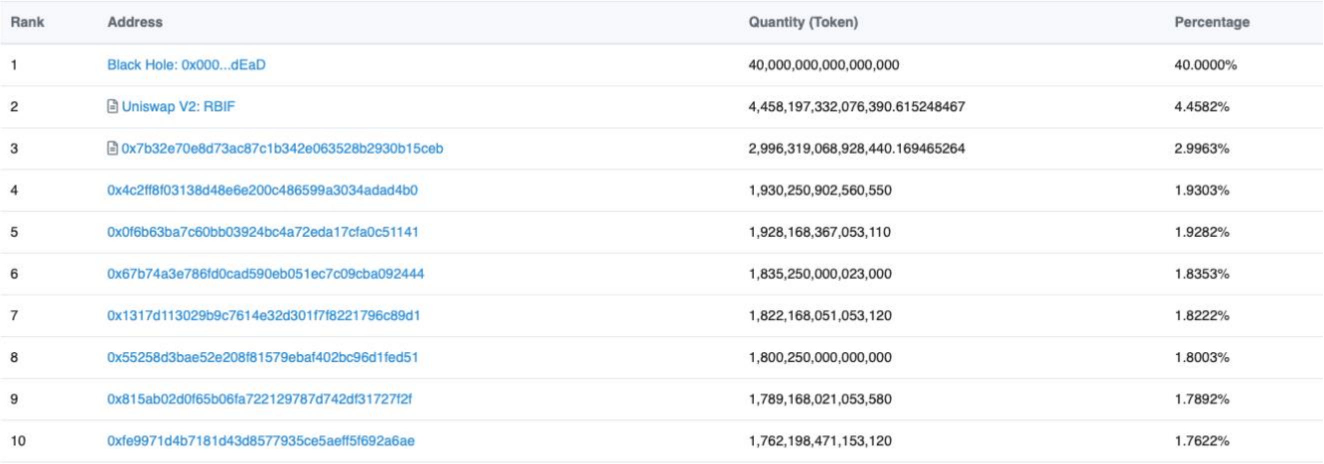
Token Contract 0x7b32e70e8d73ac87c1b342e063528b2930b15ceb (ROBO INU)

Source: Etherscan.io

Zoom: All

From Nov 6, 2021 To Nov 14, 2021

ROBO INU Top 10 Token Holders



Rank	Address	Quantity (Token)	Percentage
1	Black Hole: 0x000...dEaD	40,000,000,000,000,000	40.0000%
2	🗎 Uniswap V2: RBIF	4,458,197,332,076,390.615248467	4.4582%
3	🗎 0x7b32e70e8d73ac87c1b342e063528b2930b15ceb	2,996,319,068,928,440.169465264	2.9963%
4	0x4c2ff8f03138d48e6e200c486599a3034adad4b0	1,930,250,902,560,550	1.9303%
5	0x0f6b63ba7c60bb03924bc4a72eda17cfa0c51141	1,928,168,367,053,110	1.9282%
6	0x67b74a3e786fd0cad590eb051ec7c09cba092444	1,835,250,000,023,000	1.8353%
7	0x1317d113029b9c7614e32d301f7f8221796c89d1	1,822,168,051,053,120	1.8222%
8	0x55258d3bae52e208f81579ebaf402bc96d1fed51	1,800,250,000,000,000	1.8003%
9	0x815ab02d0f65b06fa722129787d742df31727f2f	1,789,168,021,053,580	1.7892%
10	0xfe9971d4b7181d43d8577935ce5aeff5f692a6ae	1,762,198,471,153,120	1.7622%

Contract functions details

+ [Int] IERC20
  - [Ext] totalSupply
  - [Ext] balanceOf
  - [Ext] transfer #
  - [Ext] allowance
  - [Ext] approve #
  - [Ext] transferFrom #

+ [Lib] SafeMath
  - [Int] tryAdd
  - [Int] trySub
  - [Int] tryMul
  - [Int] tryDiv
  - [Int] tryMod
  - [Int] add
  - [Int] sub
  - [Int] mul
  - [Int] div
  - [Int] mod
  - [Int] sub
  - [Int] div
  - [Int] mod

+ 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 #
  - [Int] verifyCallResult

+ Ownable (Context)
  - [Pub] <Constructor> #
  - [Pub] owner
  - [Pub] renounceOwnership #
    - modifiers: onlyOwner
  - [Pub] transferOwnership #
    - modifiers: onlyOwner
  - [Prv] _setOwner #

+ [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 #

+  Token (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] numTokensSellToAddToLiquidityAmount
  - [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 #
  - [Ext] <Fallback> ($)
  - [Prv] _reflectFee #
  - [Prv] _getValues
  - [Prv] _getTValues
  - [Prv] _getRValues
  - [Prv] _getRate
  - [Prv] _getCurrentSupply
  - [Prv] _takeLiquidity #
  - [Prv] calculateTaxFee
  - [Prv] calculateLiquidityFee
  - [Prv] calculateMarketingFee
```

  - [Prv] removeAllFee #
  - [Prv] restoreAllFee #
  - [Pub] isExcludedFromFee
  - [Pub] excludeFromFee #
    - modifiers: onlyOwner
  - [Pub] includeInFee #
    - modifiers: onlyOwner
  - [Ext] setBuyTaxFeePercent #
    - modifiers: onlyOwner
  - [Ext] setSellTaxFeePercent #
    - modifiers: onlyOwner
  - [Ext] setBuyLiquidityFeePercent #
    - modifiers: onlyOwner
  - [Ext] setSellLiquidityFeePercent #
    - modifiers: onlyOwner
  - [Ext] setMaxTxAmount #
    - modifiers: onlyOwner
  - [Ext] setBuyMarketingDivisor #
    - modifiers: onlyOwner
  - [Ext] setSellMarketingDivisor #
    - modifiers: onlyOwner
  - [Ext] setNumTokensSellToAddToLiquidity #
    - modifiers: onlyOwner
  - [Ext] setMaxTokenHolder #
    - modifiers: onlyOwner
  - [Ext] setMarketingWallet #
    - modifiers: onlyOwner
  - [Ext] burn #
  - [Int] _burnTokens #
  - [Pub] changeRouterVersion #
    - modifiers: onlyOwner
  - [Pub] setSwapAndLiquifyEnabled #
    - modifiers: onlyOwner
  - [Ext] prepareForPreSale #
    - modifiers: onlyOwner
  - [Ext] goLive #
    - 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 issue
10.	Methods execution permissions.	Passed
11.	Economy model of the contract.	High issue
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

1. Out of gas

Issue:

- The function _tokenTransfer() uses _transferStandard() function to send marketingAmt. _transferStandard operates only reflection balance, but sender address is not checked to be excluded from reward.

Recommendation:

Check the sender address to be excluded from reward and use proper functions to send marketing amount.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

2. Out of gas

Issue:

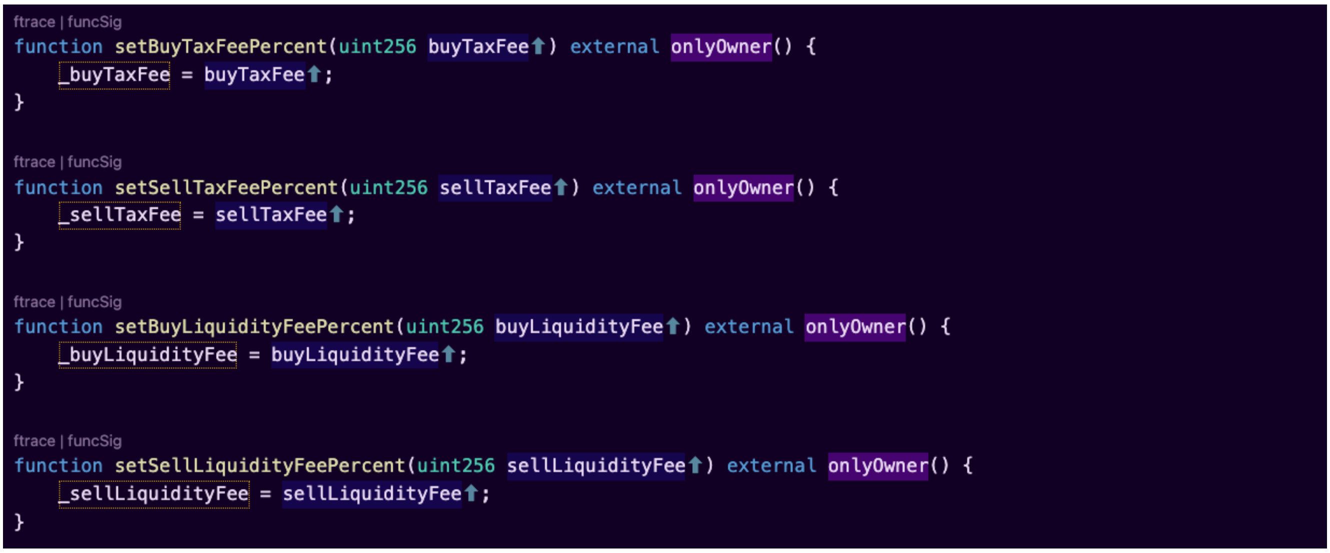
- The function _getCurrentSupply also uses the loop for evaluating total supply. It also could be aborted with OUT_OF_GAS exception if there will be a long excluded addresses list.

Recommendation:

Check that the excluded array length is not too big.

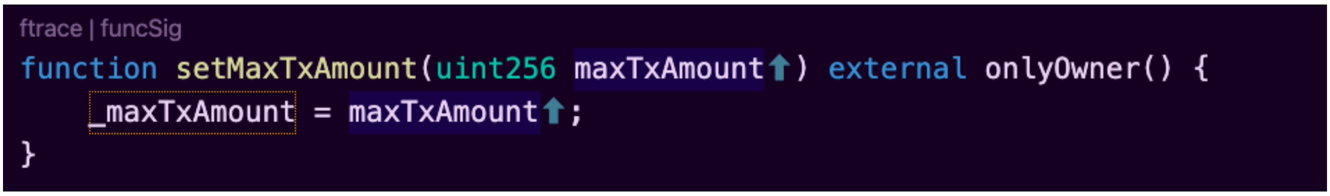
Owner privileges (In the period when the owner is not renounced)

- Owner can change buy/sell tax and liquidity fees.



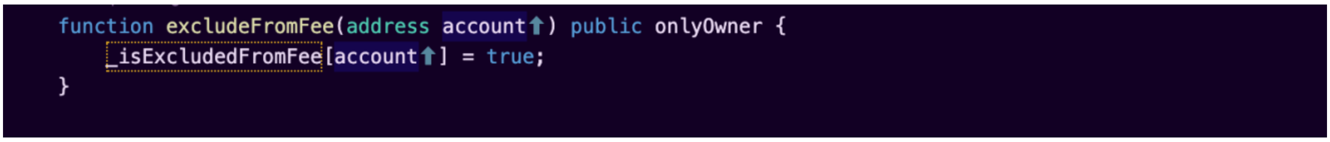
```
ftrace | funcSig
function setBuyTaxFeePercent(uint256 buyTaxFee) external onlyOwner() {
    _buyTaxFee = buyTaxFee;
}

ftrace | funcSig
function setSellTaxFeePercent(uint256 sellTaxFee) external onlyOwner() {
    _sellTaxFee = sellTaxFee;
}

ftrace | funcSig
function setBuyLiquidityFeePercent(uint256 buyLiquidityFee) external onlyOwner() {
    _buyLiquidityFee = buyLiquidityFee;
}

ftrace | funcSig
function setSellLiquidityFeePercent(uint256 sellLiquidityFee) external onlyOwner() {
    _sellLiquidityFee = sellLiquidityFee;
}
```

- Owner can change maximum transaction amount.



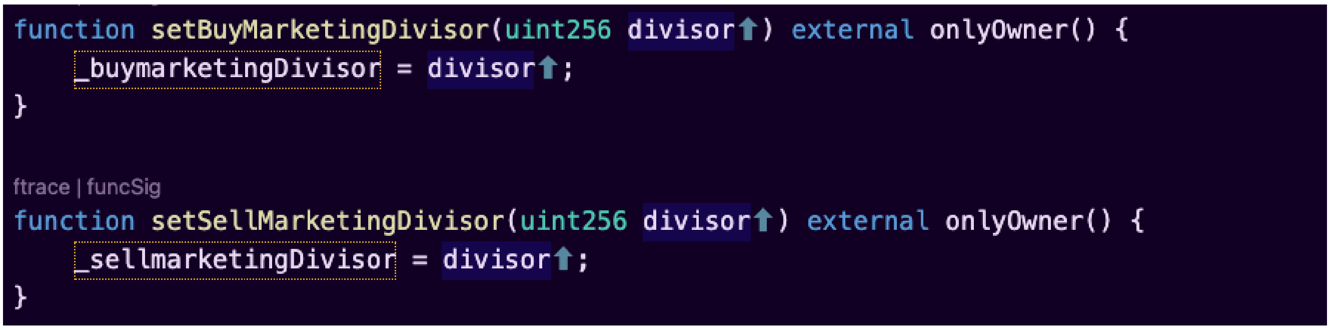
```
ftrace | funcSig
function setMaxTxAmount(uint256 maxTxAmount) external onlyOwner() {
    _maxTxAmount = maxTxAmount;
}
```

- Owner can exclude from the fee.



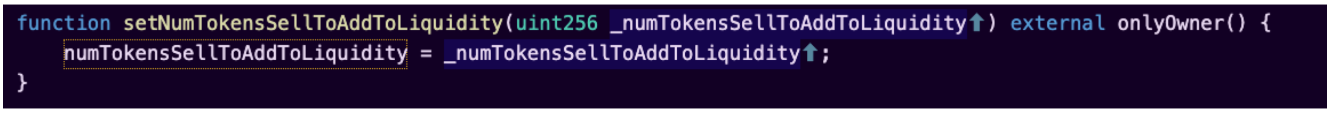
```
    function excludeFromFee(address account) public onlyOwner {
        _isExcludedFromFee[account] = true;
    }
```

- Owner can change buy/sell marketingDivisor.



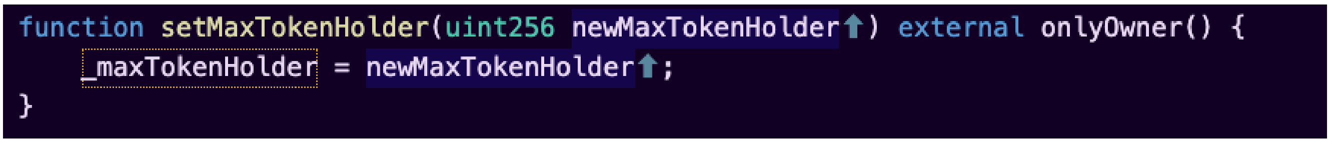
```
function setBuyMarketingDivisor(uint256 divisor) external onlyOwner() {
    _buymarketingDivisor = divisor;
}

ftrace | funcSig
function setSellMarketingDivisor(uint256 divisor) external onlyOwner() {
    _sellmarketingDivisor = divisor;
}
```

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



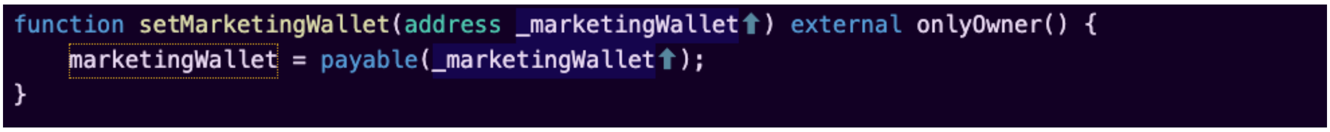
```
function setNumTokensSellToAddToLiquidity(uint256 _numTokensSellToAddToLiquidity) external onlyOwner() {
    numTokensSellToAddToLiquidity = _numTokensSellToAddToLiquidity;
}
```

- Owner can change _maxTokenHolder value.



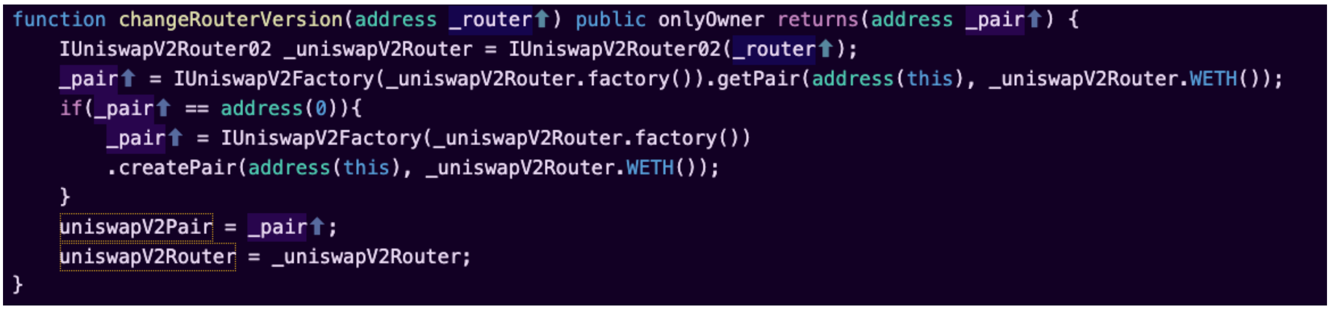
```
function setMaxTokenHolder(uint256 newMaxTokenHolder) external onlyOwner() {
    _maxTokenHolder = newMaxTokenHolder;
}
```

- Owner can change marketing address.



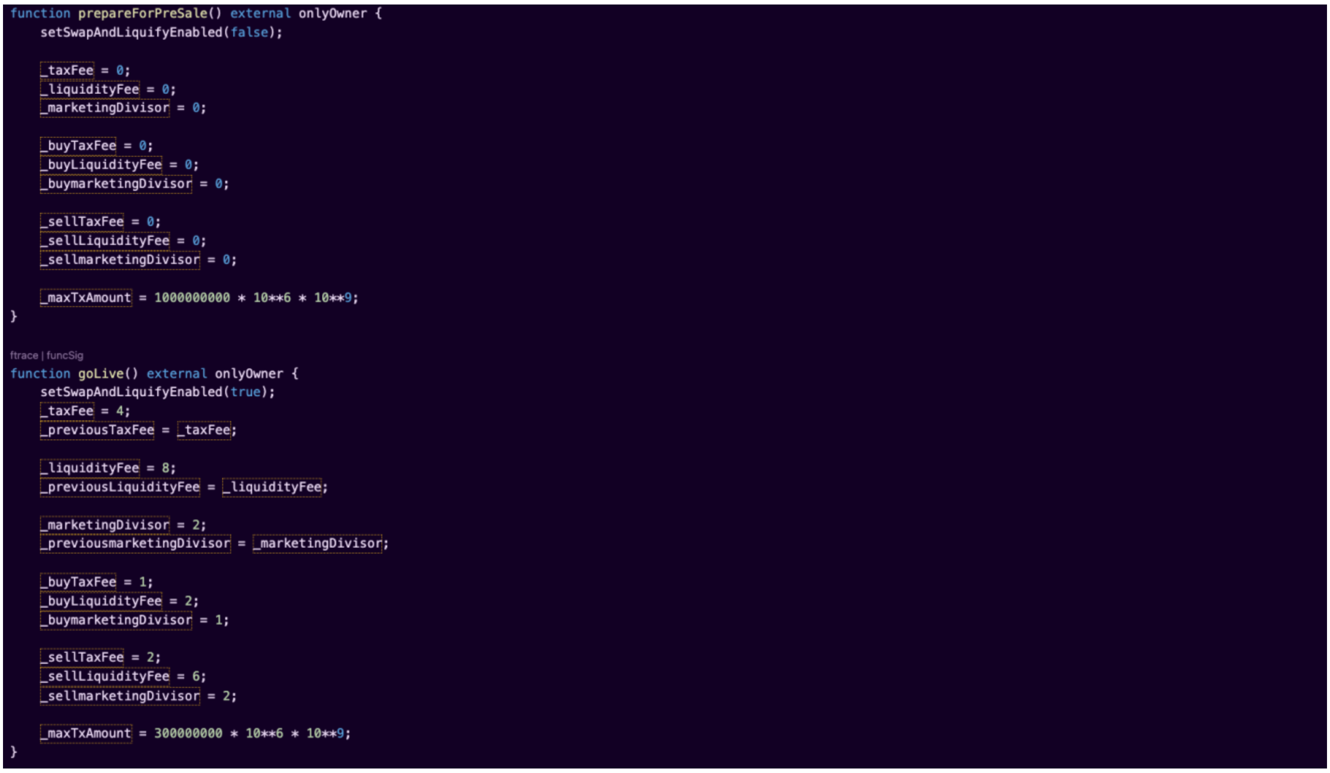
```
function setMarketingWallet(address _marketingWallet) external onlyOwner() {
    marketingWallet = payable(_marketingWallet);
}
```

- Owner can change router address.



```
function changeRouterVersion(address _router) public onlyOwner returns(address _pair) {
    IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(_router);
    _pair = IUniswapV2Factory(_uniswapV2Router.factory()).getPair(address(this), _uniswapV2Router.WETH());
    if(_pair == address(0)){
        _pair = IUniswapV2Factory(_uniswapV2Router.factory())
        .createPair(address(this), _uniswapV2Router.WETH());
    }
    uniswapV2Pair = _pair;
    uniswapV2Router = _uniswapV2Router;
}
```

- Owner can enable presale and live setting presets.



```
function prepareForPreSale() external onlyOwner {
    setSwapAndLiquifyEnabled(false);

    _taxFee = 0;
    _liquidityFee = 0;
    _marketingDivisor = 0;

    _buyTaxFee = 0;
    _buyLiquidityFee = 0;
    _buymarketingDivisor = 0;

    _sellTaxFee = 0;
    _sellLiquidityFee = 0;
    _sellmarketingDivisor = 0;

    _maxTxAmount = 1000000000 * 10**6 * 10**9;
}

ftrace | funcSig
function goLive() external onlyOwner {
    setSwapAndLiquifyEnabled(true);
    _taxFee = 4;
    _previousTaxFee = _taxFee;

    _liquidityFee = 8;
    _previousLiquidityFee = _liquidityFee;

    _marketingDivisor = 2;
    _previousmarketingDivisor = _marketingDivisor;

    _buyTaxFee = 1;
    _buyLiquidityFee = 2;
    _buymarketingDivisor = 1;

    _sellTaxFee = 2;
    _sellLiquidityFee = 6;
    _sellmarketingDivisor = 2;

    _maxTxAmount = 300000000 * 10**6 * 10**9;
}
```

Conclusion

Smart contracts contain high severity issues! Liquidity pair contract's security is not checked due to out of scope. The further transfers and operations with the funds raise are not related to this particular contract.

Liquidity locking details provided by the team:
https://app.unicrypt.network/amm/uni-v2/pair/0xa6c6b56c70abf14ed63edaffb45bb82a64454a77

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.