Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

May, 2021

Audit Details

Audited project

TomorrowWontExist

Deployer address

0x147612b77ff82418dddeF7D3B1f3146A14fCbb0e

Client contacts:

TomorrowWontExist team

Blockchain

Ethereum

Project website:

Tomorrowwontexist.com

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 TomorrowWontExist to perform an audit of smart contracts:

https://etherscan.io/address/0xb09834fa4d01c6ec44cdc530b8fa7c3e46384125#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.

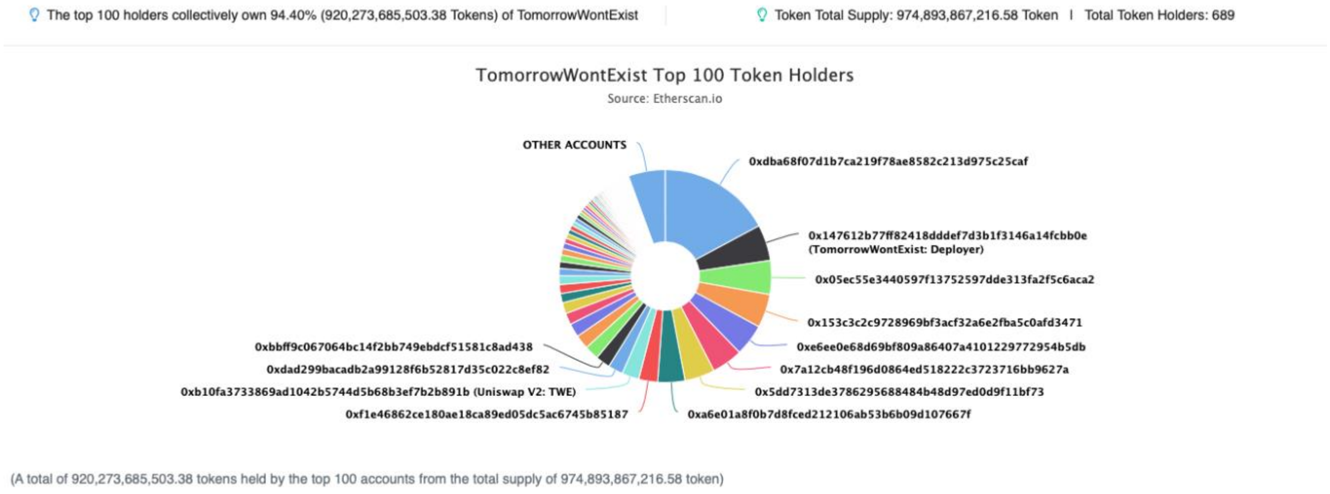
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 18.06.2021

Contract name	TomorrowWontExist
Contract address	0xb09834FA4d01C6eC44cDc530B8Fa7c3e46384125
Total supply	974,893,867,216.581050271612169833
Token ticker	TWE
Decimals	18
Token holders	689
Transactions count	4,859
Top 100 holders dominance	94.40%
Liquidity fee	3
Tax fee	3
Total tax fees	37616076884007038222201853474
Uniswap V2 pair	0xb10fa3733869ad1042b5744d5b68b3ef7b2b891b
Contract deployer address	0x147612b77ff82418dddeF7D3B1f3146A14fCbb0e
Contract’s current owner address	0x147612b77ff82418dddef7d3b1f3146a14fcbb0e

TomorrowWontExist Token Distribution



The top 100 holders collectively own 94.40% (920,273,685,503.38 Tokens) of TomorrowWontExist

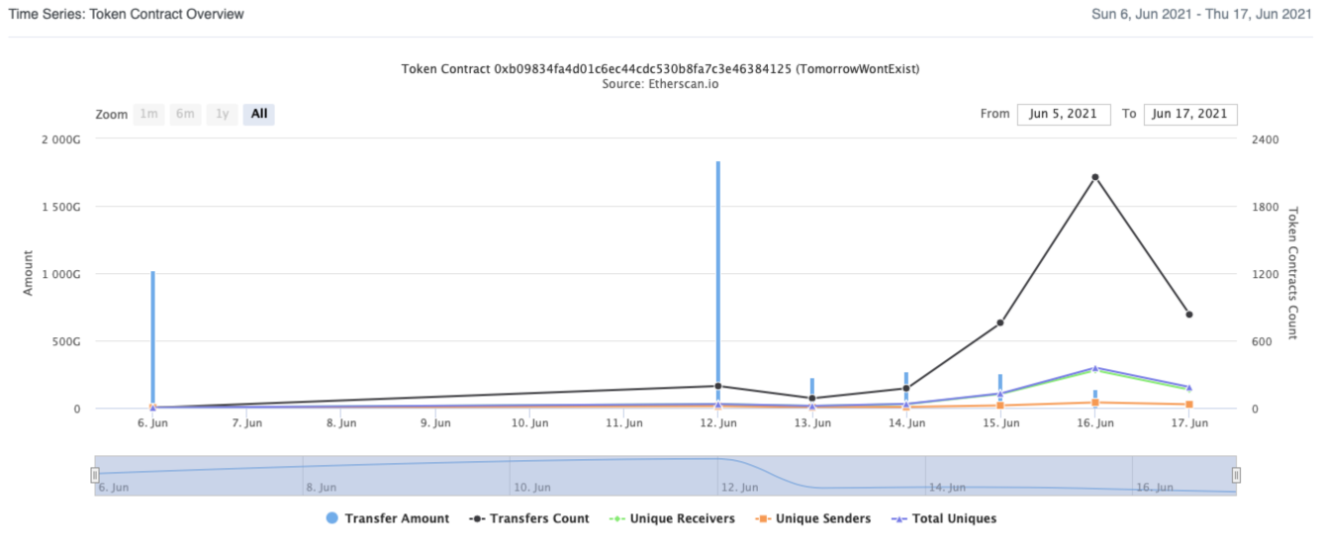
Token Total Supply: 974,893,867,216.58 Token | Total Token Holders: 689

TomorrowWontExist Top 100 Token Holders

Source: Etherscan.io

(A total of 920,273,685,503.38 tokens held by the top 100 accounts from the total supply of 974,893,867,216.58 token)

TomorrowWontExist Contract Interaction Details



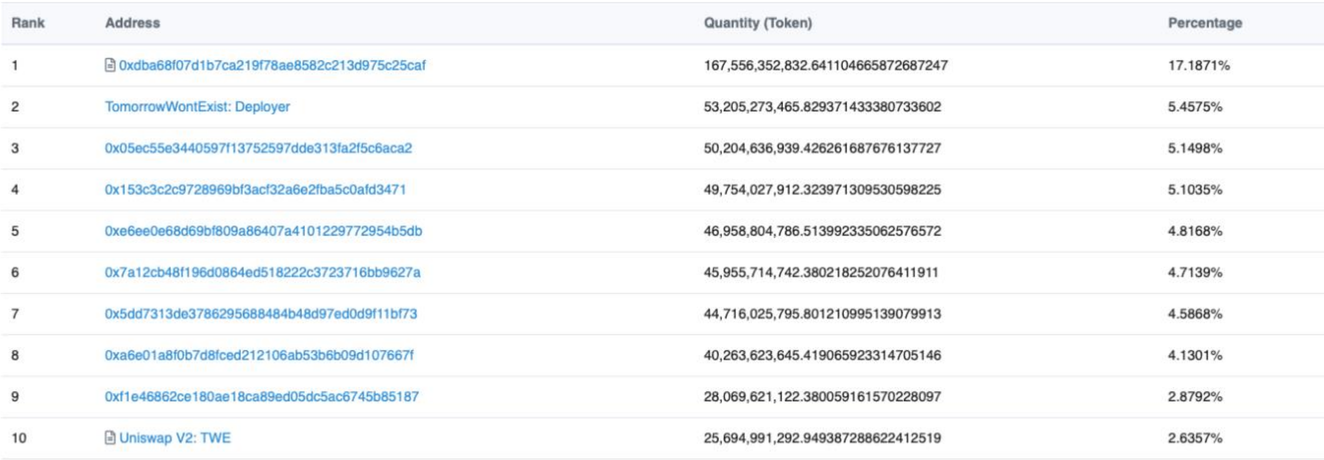
Time Series: Token Contract Overview

Sun 6, Jun 2021 - Thu 17, Jun 2021

Token Contract 0xb09834fa4d01c6ec44cdc530b8fa7c3e46384125 (TomorrowWontExist)

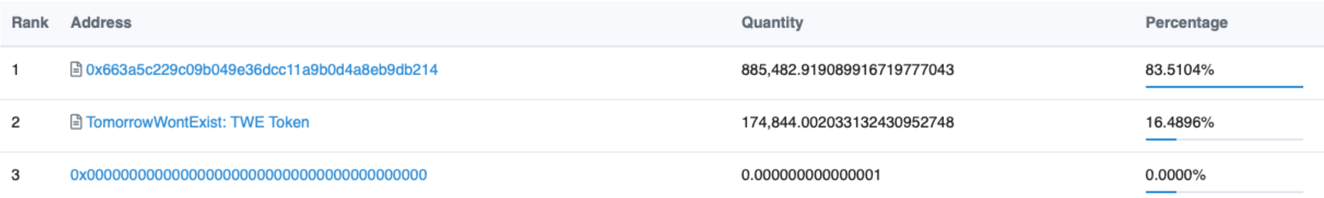
Source: Etherscan.io

TomorrowWontExist Top 10 Token Holders



Rank	Address	Quantity (Token)	Percentage
1	0xdba68f07d1b7ca219f78ae8582c213d975c25caf	167,556,352,832.641104665872687247	17.1871%
2	TomorrowWontExist: Deployer	53,205,273,465.829371433380733602	5.4575%
3	0x05ec55e3440597f13752597dde313fa2f5c6aca2	50,204,636,939.426261687676137727	5.1498%
4	0x153c3c2c9728969bf3acf32a6e2fba5c0afd3471	49,754,027,912.323971309530598225	5.1035%
5	0xe6ee0e68d69bf809a86407a4101229772954b5db	46,958,804,786.513992335062576572	4.8168%
6	0x7a12cb48f196d0864ed518222c3723716bb9627a	45,955,714,742.380218252076411911	4.7139%
7	0x5dd7313de3786295688484b48d97ed0d9f11bf73	44,716,025,795.801210995139079913	4.5868%
8	0xa6e01a8f0b7d8fced212106ab53b6b09d107667f	40,263,623,645.419065923314705146	4.1301%
9	0xf1e46862ce180ae18ca89ed05dc5ac6745b85187	28,069,621,122.380059161570228097	2.8792%
10	Uniswap V2: TWE	25,694,991,292.949387288622412519	2.6357%

TomorrowWontExist LP Token Holders



Rank	Address	Quantity	Percentage
1	0x663a5c229c09b049e36dcc11a9b0d4a8eb9db214	885,482.919089916719777043	83.5104%
2	TomorrowWontExist: TWE Token	174,844.002033132430952748	16.4896%
3	0x0000000000000000000000000000000000000000	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
  - [Int] ceil

+ [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

+ [Int] IUniswapV2Factory
  - [Ext] createPair #

+ [Int] IUniswapV2Pair
  - [Ext] sync #

+ [Int] IUniswapV2Router01
  - [Ext] factory
  - [Ext] WETH
  - [Ext] addLiquidity #
  - [Ext] addLiquidityETH ($)

```
+ [Int] IUniswapV2Router02 (IUniswapV2Router01)
  - [Ext] removeLiquidityETHSupportingFeeOnTransferTokens #
  - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #
  - [Ext] swapExactTokensForTokensSupportingFeeOnTransferTokens #
  - [Ext] swapExactETHForTokensSupportingFeeOnTransferTokens ($)

+  RewardWallet
  - [Pub] <Constructor> #

+  Balancer
  - [Pub] <Constructor> #

+  TomorrowWontExist (Context, IERC20, Ownable)
  - [Pub] <Constructor> #
  - [Pub] name
  - [Pub] symbol
  - [Pub] decimals
  - [Pub] totalSupply
  - [Int] find2Percent
  - [Pub] balanceOf
  - [Pub] transfer #
  - [Pub] allowance
  - [Pub] approve #
  - [Pub] transferFrom #
  - [Pub] increaseAllowance #
  - [Pub] decreaseAllowance #
  - [Pub] isExcluded
  - [Pub] reflectionFromToken
  - [Pub] tokenFromReflection
  - [Ext] excludeAccount #
    - modifiers: onlyOwner
  - [Ext] includeAccount #
    - modifiers: onlyOwner
  - [Prv] _approve #
  - [Prv] _transfer #
  - [Prv] collectFee #
  - [Prv] _getReflectionRate
  - [Prv] swapAndLiquify #
    - modifiers: lockTheSwap
  - [Prv] swapTokensForEth #
  - [Prv] addLiquidity #
  - [Ext] setPair #
    - modifiers: onlyOwner
  - [Ext] setTaxless #
    - modifiers: onlyOwner
  - [Ext] setSwapAndLiquifyEnabled #
    - modifiers: onlyOwner
  - [Ext] setFeeActive #
    - modifiers: onlyOwner
  - [Ext] setTaxFee #
    - modifiers: onlyOwner
  - [Ext] setBurnFee #
    - modifiers: onlyOwner
  - [Ext] setLiquidityFee #
    - modifiers: onlyOwner
```

- [Ext] setDev #
  - modifiers: onlyOwner
- [Ext] setMaxTxAmount #
  - modifiers: onlyOwner
- [Ext] setMinTokensBeforeSwap #
  - modifiers: onlyOwner
- [Ext] <Fallback> ($)

($) = 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

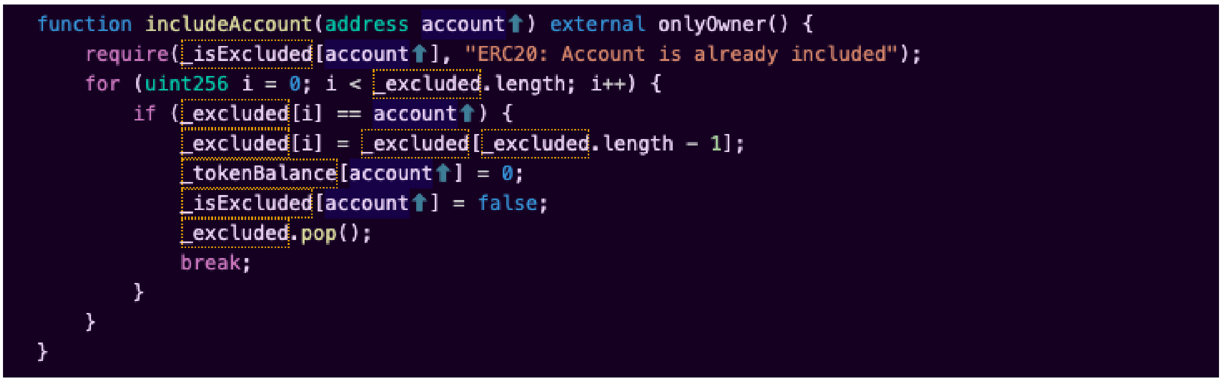
No medium severity issues found.

Low Severity Issues

1. Out of gas

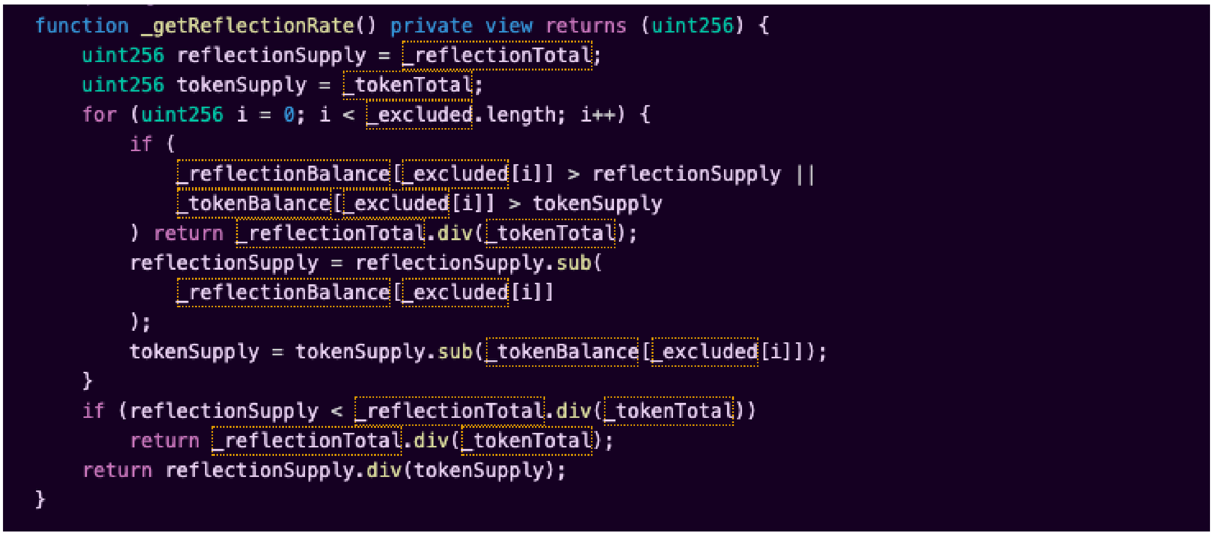
Issue:

- The function includeAccount () 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 includeAccount(address account) external onlyOwner() {
    require(_isExcluded[account], "ERC20: Account is already included");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account) {
            _excluded[i] = _excluded[_excluded.length - 1];
            _tokenBalance[account] = 0;
            _isExcluded[account] = false;
            _excluded.pop();
            break;
        }
    }
}
```

- The function _getReflectionRate() 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 _getReflectionRate() private view returns (uint256) {
    uint256 reflectionSupply = _reflectionTotal;
    uint256 tokenSupply = _tokenTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _reflectionBalance[_excluded[i]] > reflectionSupply ||
            _tokenBalance[_excluded[i]] > tokenSupply
        ) return _reflectionTotal.div(_tokenTotal);
        reflectionSupply = reflectionSupply.sub(
            _reflectionBalance[_excluded[i]]
        );
        tokenSupply = tokenSupply.sub(_tokenBalance[_excluded[i]]);
    }
    if (reflectionSupply < _reflectionTotal.div(_tokenTotal))
        return _reflectionTotal.div(_tokenTotal);
    return reflectionSupply.div(tokenSupply);
}
```

Recommendation:

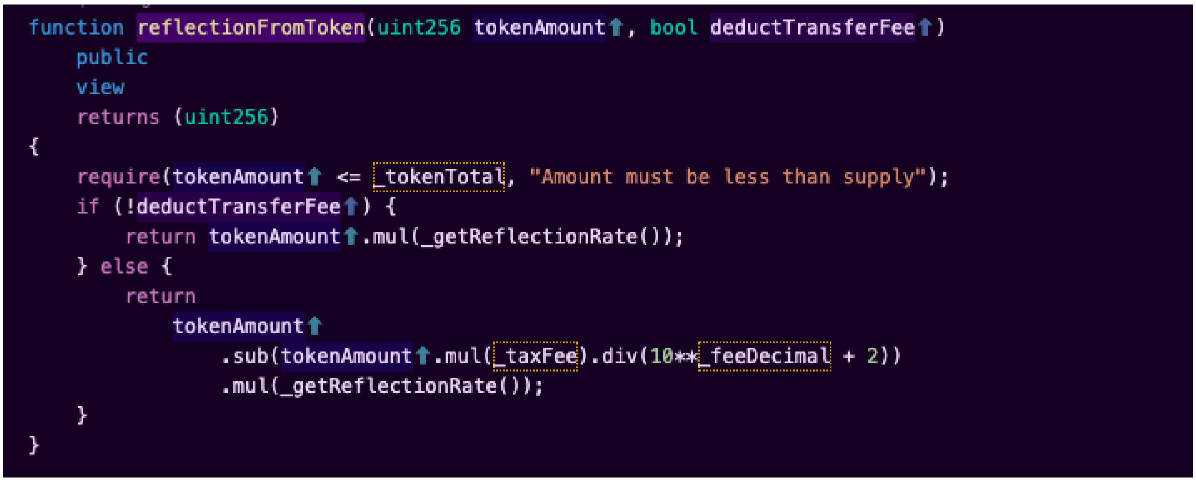
Check that the excluded array length is not too big.

2. Wrong reflection from token calculations

Issue:

- Missing parentheses when calculating target value.

tokenAmount
*.sub(tokenAmount.mul(_taxFee).div(10**(_feeDecimal + 2)))*
.mul(_getReflectionRate());



```
function reflectionFromToken(uint256 tokenAmount, bool deductTransferFee)
    public
    view
    returns (uint256)
{
    require(tokenAmount <= _tokenTotal, "Amount must be less than supply");
    if (!deductTransferFee) {
        return tokenAmount.mul(_getReflectionRate());
    } else {
        return
            tokenAmount
                .sub(tokenAmount.mul(_taxFee).div(10**_feeDecimal + 2))
                .mul(_getReflectionRate());
    }
}
```

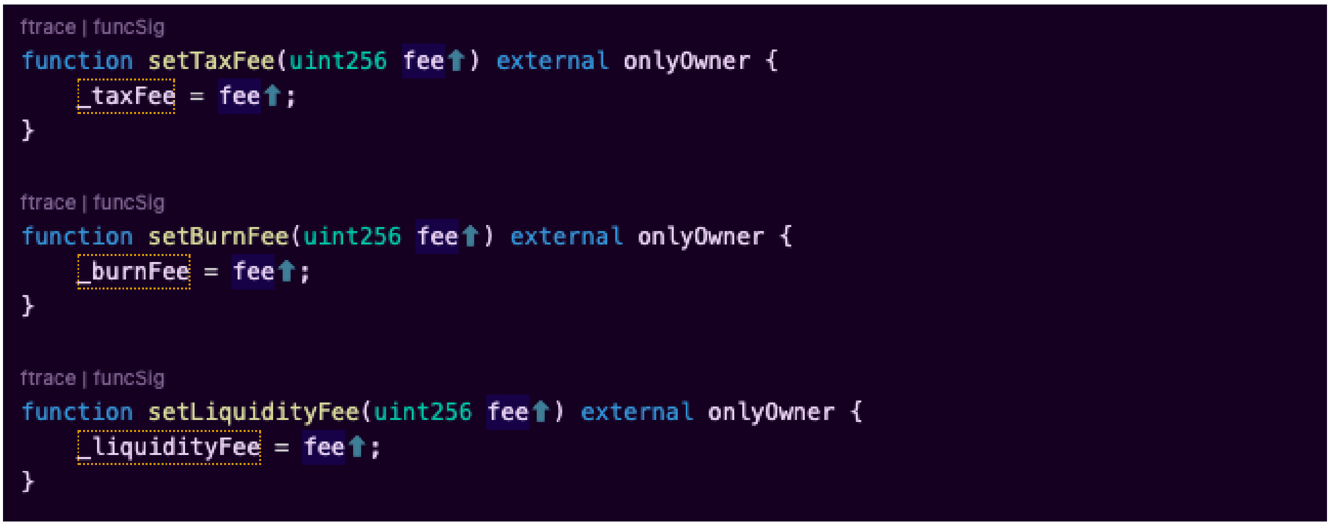
3. No checking if dev address is excluded

Issue:

- There is no checking if dev address is excluded from reward in _transfer function, so if it would be, token balance of dev address won’t increase.

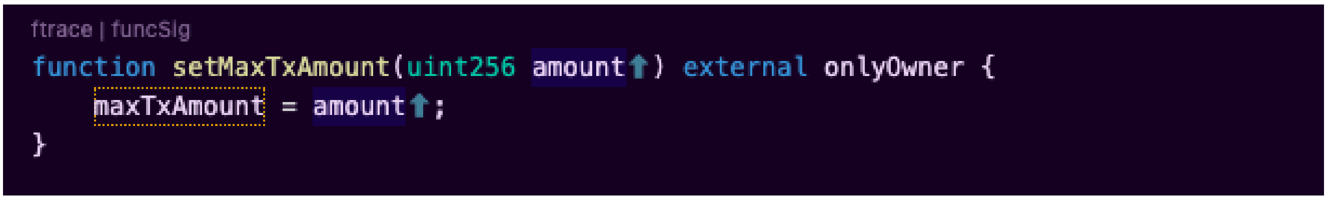
Owner privileges (In the period when the owner is not renounced)

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



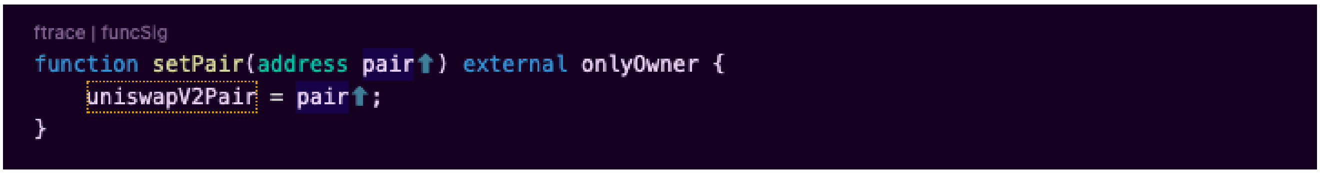
```
ftrace | funcSig
function setTaxFee(uint256 fee) external onlyOwner {
    _taxFee = fee;
}

ftrace | funcSig
function setBurnFee(uint256 fee) external onlyOwner {
    _burnFee = fee;
}

ftrace | funcSig
function setLiquidityFee(uint256 fee) external onlyOwner {
    _liquidityFee = fee;
}
```

- Owner can change the maximum transaction amount.



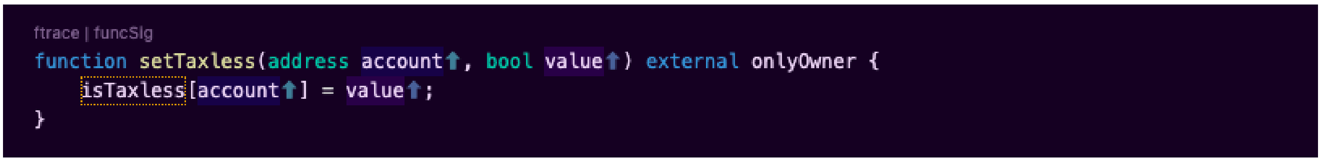
```
ftrace | funcSig
function setMaxTxAmount(uint256 amount) external onlyOwner {
    maxTxAmount = amount;
}
```

- Owner can change uniswapV2Pair.



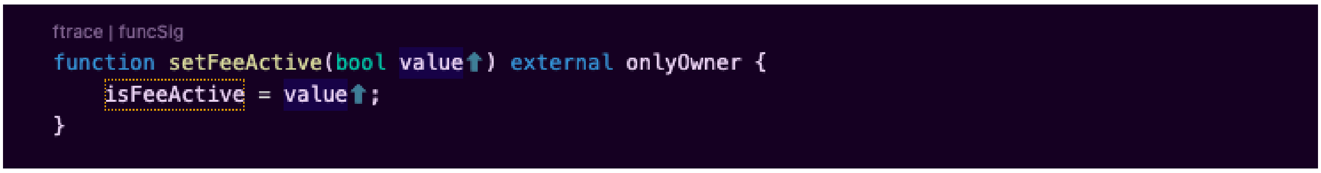
```
ftrace | funcSig
function setPair(address pair) external onlyOwner {
    uniswapV2Pair = pair;
}
```

- Owner can exclude from the taxes.



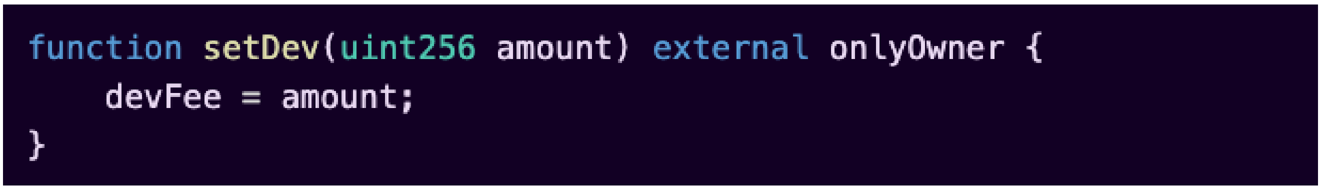
```
ftrace | funcSig
function setTaxless(address account, bool value) external onlyOwner {
    isTaxless[account] = value;
}
```

- Owner can disable and enable fees.



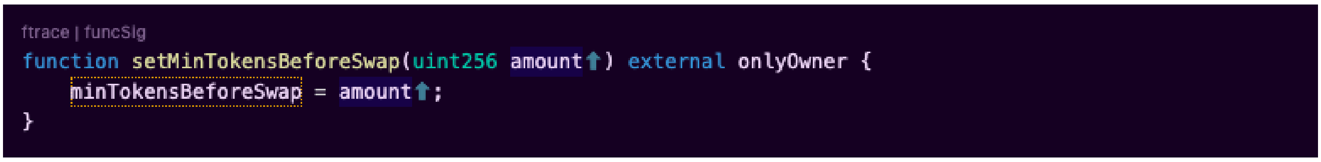
```
ftrace | funcSig
function setFeeActive(bool value) external onlyOwner {
    isFeeActive = value;
}
```

- Owner can change dev fee.



```
function setDev(uint256 amount) external onlyOwner {
    devFee = amount;
}
```

- Owner can change minimum amount of tokens needed to swap.



```
ftrace | funcSig
function setMinTokensBeforeSwap(uint256 amount) external onlyOwner {
    minTokensBeforeSwap = amount;
}
```

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://app.unicrypt.network/amm/uni-v2/pair/0xb10fa3733869ad1042b5744d5b68b3ef7b2b891b

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.