Techrate1 Techrate

TechRate

AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details

Audited project

NarakaToken

Deployer address

0x4d92eb557dc0c930c5c8f87160181d78dc25d78e

Client contacts:

NarakaToken team

Blockchain

Ethereum

Project website:

https://NarakaToken.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 NarakaToken to perform an audit of smart contracts:
https://etherscan.io/address/0x8e3fe7cdf4ebb605bbbac3a43d76ea757f7f06e2#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.

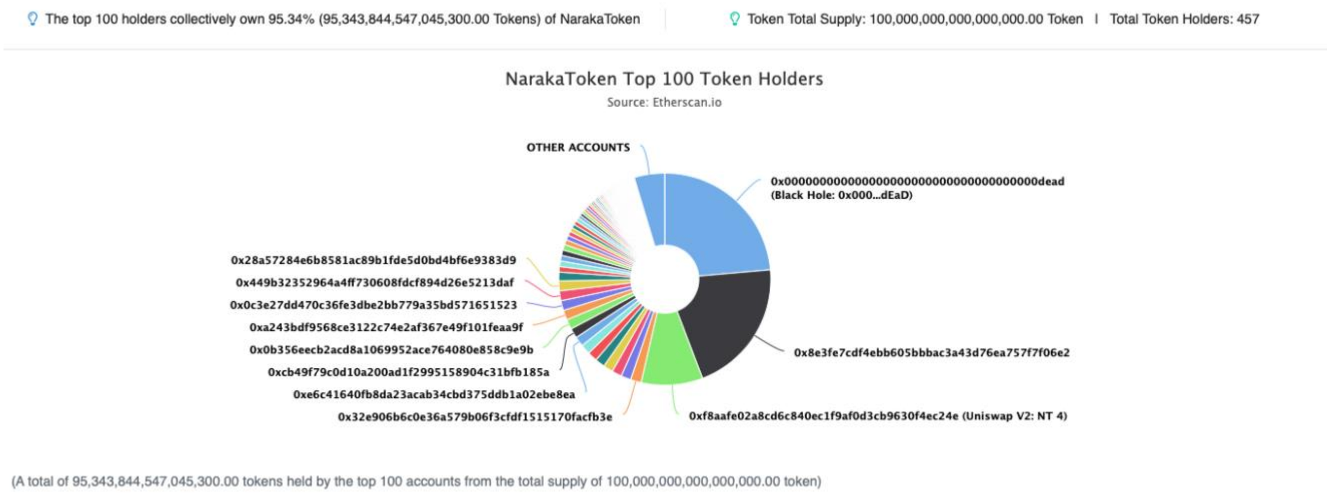
The information in this report should be used to understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contracts Details

Token contract details for 26.11.2021

Contract name	NarakaToken
Contract address	0x8e3fE7cDF4eBB605bBbac3a43d76Ea757F7F06e2
Total supply	100,000,000,000,000,000
Token ticker	NT
Decimals	9
Token holders	457
Transactions count	1,217
Top 100 holders dominance	95.34%
Buy/Sell marketing fee	70/70
Buy/Sell reflection fee	10/10
Total fees	924267061405828470917323
Uniswap V2 pair	0xf8aafe02a8cd6c840ec1f9af0d3cb9630f4ec24e
Contract deployer address	0x4d92eb557dc0c930c5c8f87160181d78dc25d78e
Contract's current owner address	0x4d92eb557dc0c930c5c8f87160181d78dc25d78e

NarakaToken Token Distribution


The top 100 holders collectively own 95.34% (95,343,844,547,045,300.00 Tokens) of NarakaToken

Token Total Supply: 100,000,000,000,000,000.00 Token | Total Token Holders: 457

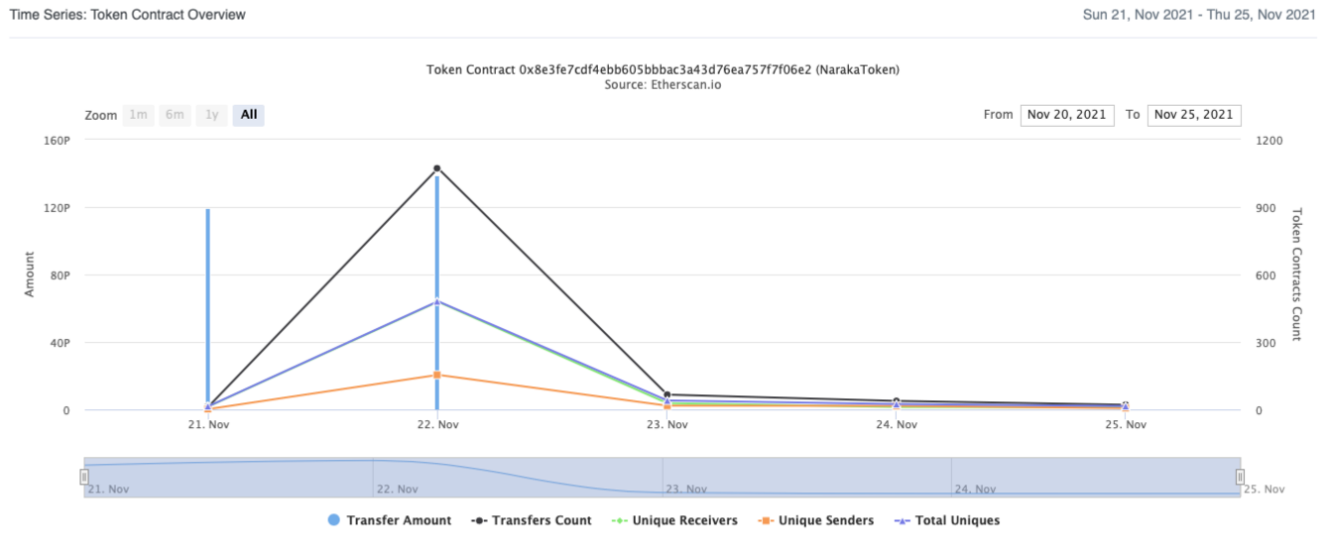
NarakaToken Top 100 Token Holders

Source: Etherscan.io

OTHER ACCOUNTS
0x000000000000000000000000000000000000dead (Black Hole: 0x000...dEaD)
0x8e3fe7cdf4ebb605bbbac3a43d76ea757f7f06e2
0xf8aafe02a8cd6c840ec1f9af0d3cb9630f4ec24e (Uniswap V2: NT 4)
0x28a57284e6b8581ac89b1fde5d0bd4bf6e9383d9
0x449b32352964a4ff730608fdcf894d26e5213daf
0x0c3e27dd470c36fe3dbe2bb779a35bd571651523
0xa243bdf9568ce3122c74e2af367e49f101feaa9f
0x0b356eecb2acd8a1069952ace764080e858c9e9b
0xcb49f79c0d10a200ad1f2995158904c31bfb185a
0xe6c41640fb8da23acab34cbd375ddb1a02ebe8ea
0x32e906b6c0e36a579b06f3cfdf1515170facfb3e

(A total of 95,343,844,547,045,300.00 tokens held by the top 100 accounts from the total supply of 100,000,000,000,000,000.00 token)

NarakaToken Contract Interaction Details

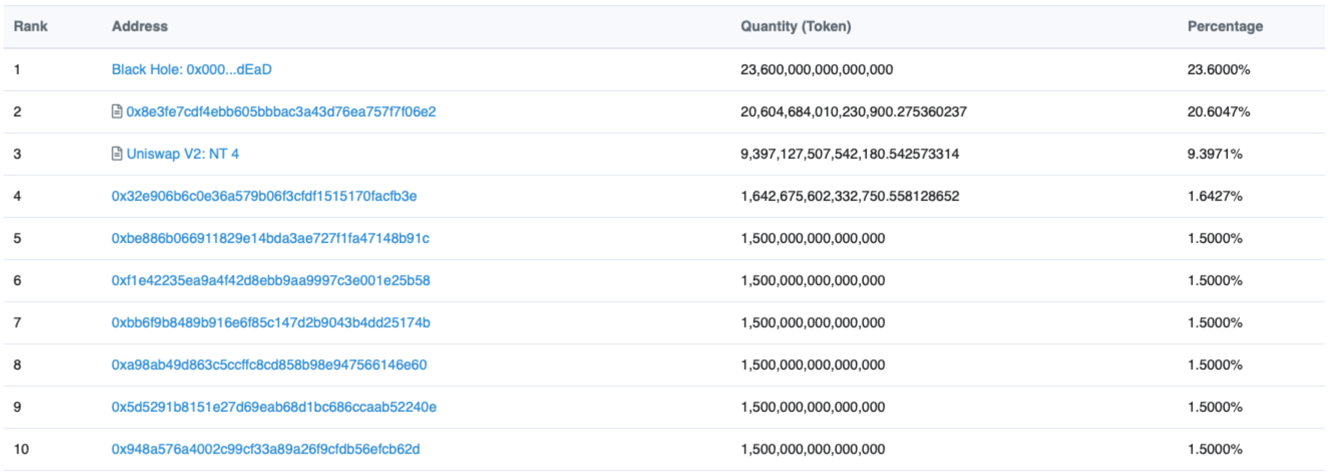

Time Series: Token Contract Overview

Sun 21, Nov 2021 - Thu 25, Nov 2021

Token Contract 0x8e3fe7cdf4ebb605bbbac3a43d76ea757f7f06e2 (NarakaToken)

Source: Etherscan.io

NarakaToken Top 10 Token Holders



Rank	Address	Quantity (Token)	Percentage
1	Black Hole: 0x000...dEaD	23,600,000,000,000,000	23.6000%
2	0x8e3fe7cdf4ebb605bbbac3a43d76ea757f7f06e2	20,604,684,010,230,900.275360237	20.6047%
3	Uniswap V2: NT 4	9,397,127,507,542,180.542573314	9.3971%
4	0x32e906b6c0e36a579b06f3cfdf1515170facfb3e	1,642,675,602,332,750.558128652	1.6427%
5	0xbe886b066911829e14bda3ae727f1fa47148b91c	1,500,000,000,000,000	1.5000%
6	0xf1e42235ea9a4f42d8ebb9aa9997c3e001e25b58	1,500,000,000,000,000	1.5000%
7	0xbb6f9b8489b916e6f85c147d2b9043b4dd25174b	1,500,000,000,000,000	1.5000%
8	0xa98ab49d863c5ccffc8cd858b98e947566146e60	1,500,000,000,000,000	1.5000%
9	0x5d5291b8151e27d69eab68d1bc686ccaab52240e	1,500,000,000,000,000	1.5000%
10	0x948a576a4002c99cf33a89a26f9cfdb56efcb62d	1,500,000,000,000,000	1.5000%

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

+ [Lib] Address
- [Int] isContract
- [Int] sendValue #
- [Int] functionCall #
- [Int] functionCall #
- [Int] functionCallWithValue #
- [Int] functionCallWithValue #
- [Prv] _functionCallWithValue #

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

+ [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] 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 #

+ NarakaToken (Context, IERC20, Ownable)

- [Pub] <Constructor> #
- [Ext] openTrading #
  - modifiers: onlyOwner
- [Ext] setZeroBuyTaxmode #
  - modifiers: onlyOwner
- [Ext] setAntiBotmode #
  - modifiers: onlyOwner
- [Ext] setNewRouter #
  - 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] _approve #
- [Prv] _transfer #
- [Prv] swapTokens #
  - modifiers: lockTheSwap
- [Prv] sendETHToFee #
- [Prv] swapTokensForEth #
- [Prv] addLiquidity #
- [Prv] _tokenTransfer #
- [Prv] _transferStandard #
- [Prv] _transferToExcluded #
- [Prv] _transferFromExcluded #
- [Prv] _transferBothExcluded #
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _takeLiquidity #
- [Prv] calculateTaxFee
- [Prv] calculateLiquidityFee
- [Pub] excludeMultiple #
  - modifiers: onlyOwner
- [Pub] excludeFromFee #
  - modifiers: onlyOwner
- [Pub] includeInFee #

  - modifiers: onlyOwner
- [Ext] setWallets #
  - modifiers: onlyOwner
- [Prv] transferToAddressETH #
- [Pub] isSniper
- [Pub] manage_Snipers #
  - modifiers: onlyOwner
- [Pub] manage_trusted #
  - modifiers: onlyOwner
- [Pub] withDrawLeftoverETH #
  - modifiers: onlyOwner
- [Pub] withdrawStuckTokens #
  - modifiers: onlyOwner
- [Ext] setMaxWalletPercent_base1000 #
  - modifiers: onlyOwner
- [Ext] setMaxWalletExempt #
  - modifiers: onlyOwner
- [Ext] setSwapSettings #
  - modifiers: onlyOwner
- [Ext] multiTransfer #
  - modifiers: onlyOwner
- [Ext] multiTransfer_fixed #
  - modifiers: onlyOwner
- [Ext] setTaxesBuy #
  - modifiers: onlyOwner
- [Ext] setTaxesSell #
  - 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

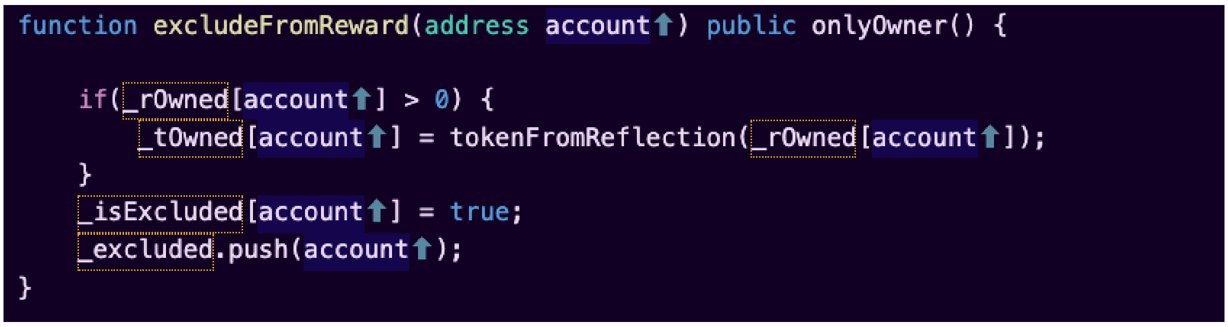
No medium severity issues found.

Low Severity Issues

1. Exlude from reward

Issue:

- The function excludeFromReward() do not check address to be already excluded.



```
function excludeFromReward(address account) public onlyOwner() {

    if(_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}
```

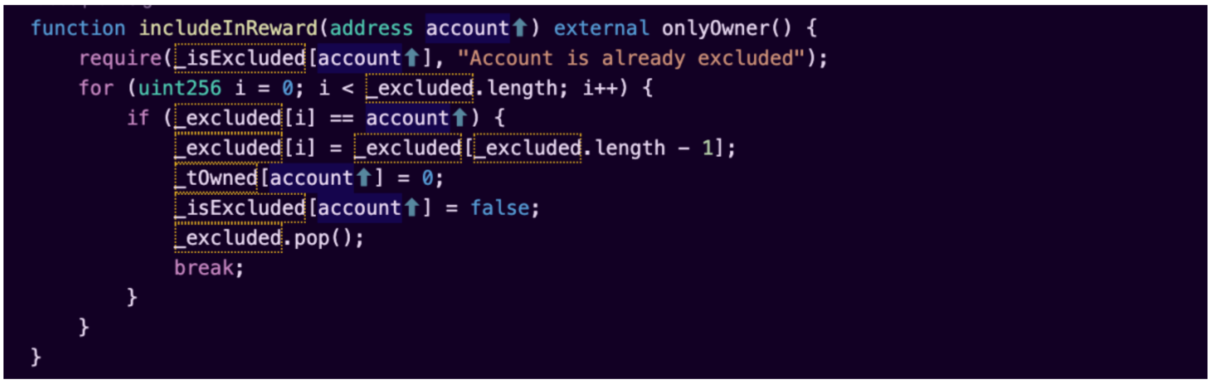
Recommendation:

Check that the addresses to be excluded is not already excluded.

2. Out of gas

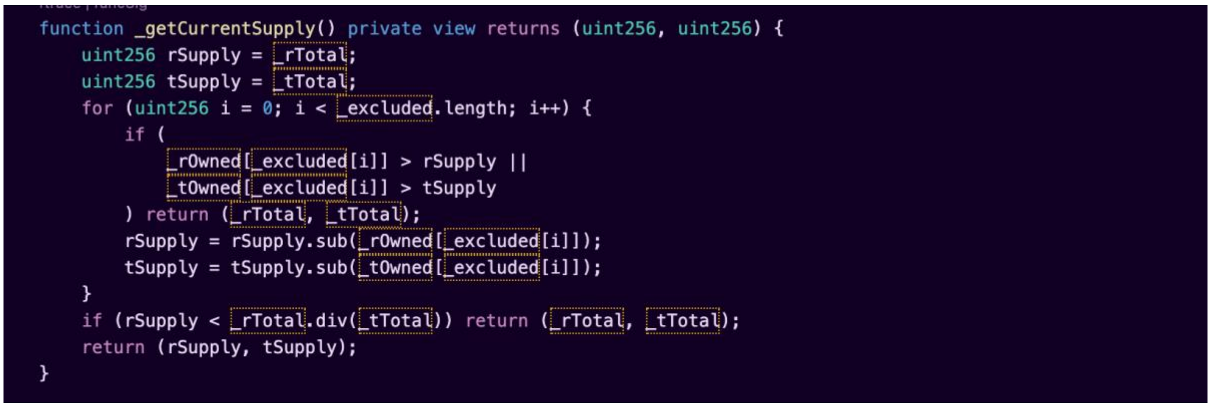
Issue:

- 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.

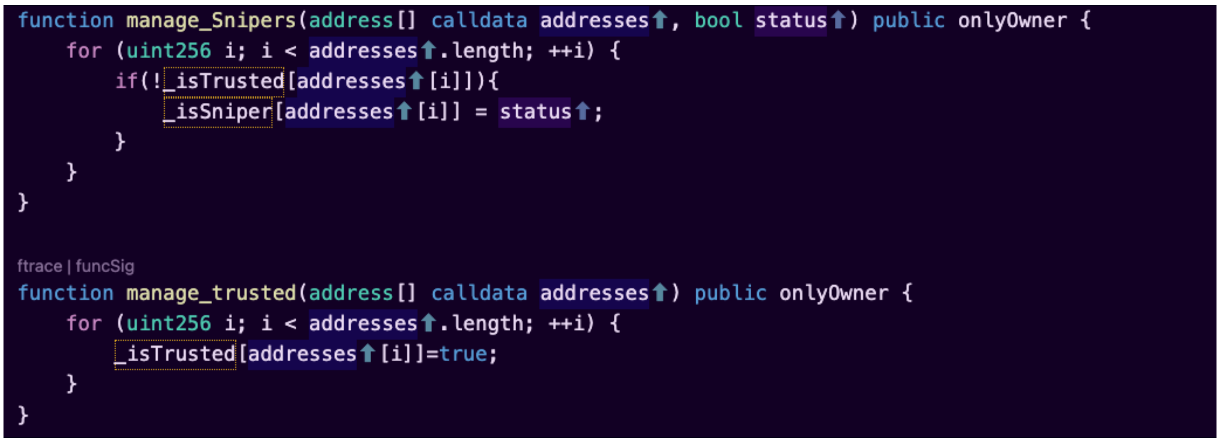


```
function _getCurrentSupply() private view returns (uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (
            _rOwned[_excluded[i]] > rSupply ||
            _tOwned[_excluded[i]] > tSupply
      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

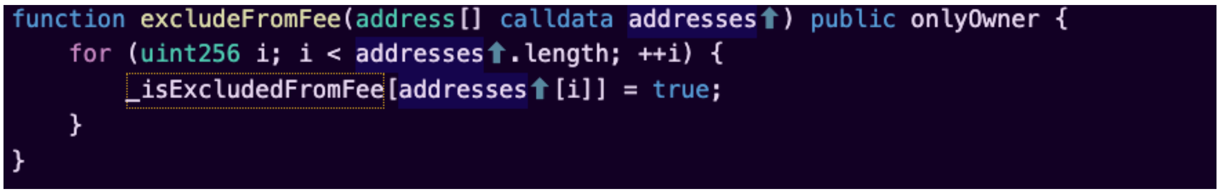
- The function manage_trusted() and manage_Snipers() uses the loop for iterating through addresses list from the function argument. It also could be aborted with OUT_OF_GAS exception if there will be a long addresses list.



```
function manage_Snipers(address[] calldata addresses, bool status) public onlyOwner {
    for (uint256 i; i < addresses.length; ++i) {
        if(!_isTrusted[addresses[i]]){
            _isSniper[addresses[i]] = status;
        }
    }
}

ftrace | funcSig
function manage_trusted(address[] calldata addresses) public onlyOwner {
    for (uint256 i; i < addresses.length; ++i) {
        _isTrusted[addresses[i]]=true;
    }
}
```

- The function excludeFromFee() uses the loop for iterating through addresses list from the function argument. It also could be aborted with OUT_OF_GAS exception if there will be a long addresses list.



```
function excludeFromFee(address[] calldata addresses) public onlyOwner {
    for (uint256 i; i < addresses.length; ++i) {
        _isExcludedFromFee[addresses[i]] = true;
    }
}
```

Recommendation:

Check that the array length is not too big.

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

- Owner can enable trading.
- Owner can enable/disable zeroBuyTaxmode.
- Owner can enable/disable anti bot mode.
- Owner can change router address.
- Owner can exclude from fees.
- Owner can change marketing and dev wallets.
- Owner can change adresses sniper status.
- Owner can withdraw contract BNBs and tokens.
- Owner can change max wallet token.
- Owner can exclude from max wallet token.
- Owner can change swapThreshold.
- Owner can change fees.
- Owner can run multiple transfer.

Conclusion

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

Liquidity locking details provided by the team:
https://www.team.finance/view-coin/0x8e3fE7cDF4eBB605bBbac3a43d76Ea757F7F06e2?name=NarakaToken&symbol=NT

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.