TechRate

AUDIT COMPANY

Smart Contract Security Audit

Audit Details

Audited project

Meo.tools

Deployer address

0xdac833a4fbb2f7173c498646981774547c17d43a

Client contacts:

Meo.tools team

Blockchain

Binance Smart Chain

Project website:

https://meo.tools

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 Meo.tools to perform an audit of smart contracts:

https://bscscan.com/address/0x37c4a3a9acfaa4bbe543dcb0d6ce4b2443b3b0d7#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 04.09.2021

|	
Contract name	Meo.tools
Contract address	0x37C4A3a9ACFaa4BBe543DCB0D6CE4b2443B3b0D7
Total supply	1,000,000,000
Token ticker	MEO
Decimals	9
Token holders	1
Transactions count	1
Top 100 holders dominance	100.00%
Research fee	2
Dev fee	3
Liquidity fee	3
Rfi fee	2
Total fees	0
Pancake V2 pair	0x60d10826643ead62c8c26f8270cb9a8fd0cbbed2
Contract deployer address	0xdac833a4fbb2f7173c498646981774547c17d43a
Contract's current owner address	0xdac833a4fbb2f7173c498646981774547c17d43a

Meo.tools Token Distribution

Meo.tools Top 100 Token Holders
Source: BscScan.com

OTHER ACCOUNTS

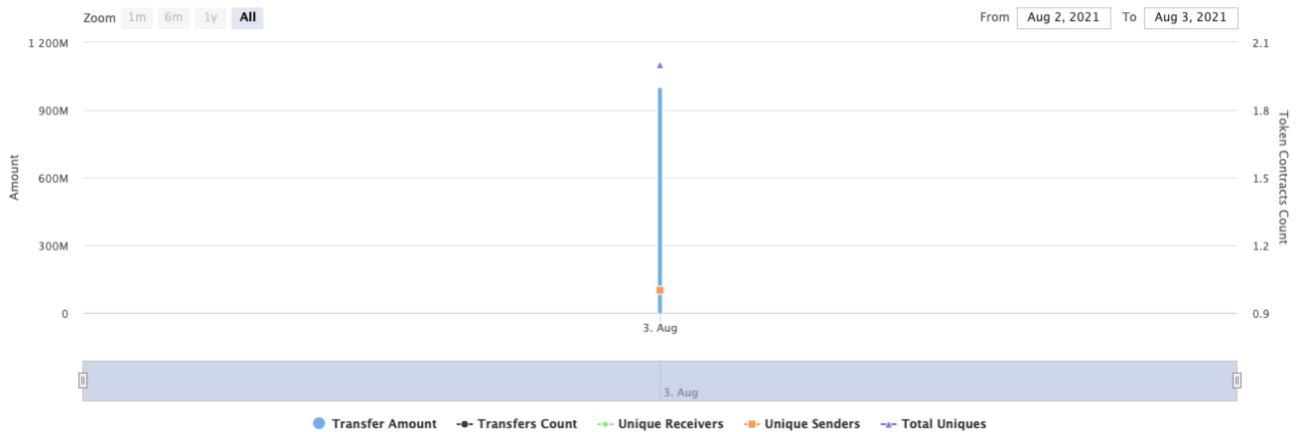0xdac833a4fbb2f7173c498646981774547c17d43a
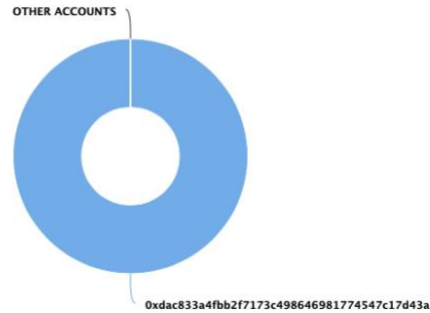
(A total of 1,000,000,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000.00 token)

Meo.tools Contract Interaction Details

Time Series: Token Contract Overview　　　　　　　　　　　　　　Tue 3, Aug 2021 - Tue 3, Aug 2021

Token Contract 0x37c4a3a9acfaa4bbe543dcb0d6ce4b2443b3b0d7 (Meo.tools)
Source: BscScan.com

Zoom 1m 6m 1y All　　　　　　　　　　　　　　　　From Aug 2, 2021 To Aug 3, 2021

● Transfer Amount　　-●- Transfers Count　　-●- Unique Receivers　　-■- Unique Senders　　-▲- Total Uniques

Meo.tools Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0xdac833a4fbb2f7173c498646981774547c17d43a	1,000,000,000	100.0000%

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 **#**
- **[Prv]** _verifyCallResult

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

+ [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 **#**

+ **[Int]** IUniswapV2Factory
  - **[Ext]** feeTo
  - **[Ext]** feeToSetter
  - **[Ext]** getPair
  - **[Ext]** allPairs
  - **[Ext]** allPairsLength
  - **[Ext]** createPair **#**
  - **[Ext]** setFeeTo **#**
  - **[Ext]** setFeeToSetter **#**

+ **MEO (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]** reflectionFromToken
  - **[Pub]** tokenFromReflection
  - **[Pub]** excludeFromRFI **#**
    - modifiers: onlyOwner
  - **[Ext]** includeInRFI **#**
    - modifiers: onlyOwner
  - **[Pub]** excludeFromFeeAndRfi **#**

- modifiers: onlyOwner
- **[Pub]** excludeFromFee **#**
  - modifiers: onlyOwner
- **[Pub]** includeInFee **#**
  - modifiers: onlyOwner
- **[Pub]** isExcludedFromFee
- **[Pub]** setRfiRatesPercents **#**
  - modifiers: onlyOwner
- **[Pub]** setWallets **#**
  - modifiers: onlyOwner
- **[Pub]** setPresaleWallet **#**
  - modifiers: onlyOwner
- **[Ext]** setMaxTxPercent **#**
  - modifiers: onlyOwner
- **[Ext]** setMaxTxAmount **#**
  - modifiers: onlyOwner
- **[Ext]** setThreshholdForLP **#**
  - modifiers: onlyOwner
- **[Pub]** setSwapAndLiquifyEnabled **#**
  - modifiers: onlyOwner
- **[Ext]** **<Fallback>** **($)**
- **[Prv]** _reflectRfi **#**
- **[Prv]** _getValues
- **[Prv]** _getTValues
- **[Prv]** _getRValues
- **[Prv]** _getRate
- **[Prv]** _getCurrentSupply
- **[Prv]** _takeLiquidity **#**
- **[Prv]** _approve **#**
- **[Prv]** _transfer **#**
- **[Prv]** _tokenTransfer **#**
- **[Prv]** reflectDevandResearchFee **#**
- **[Prv]** swapAndLiquify **#**
  - modifiers: lockTheSwap
- **[Prv]** swapTokensForBNB **#**
- **[Prv]** addLiquidity **#**
- **[Pub]** totalDevelopmentFee
- **[Pub]** totalResearchFee

($) = 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.	Low issues
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 includeInRFI() 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.

```solidity
function includeInRFI(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is not 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.

```solidity
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.

2. Wrong reflectDevandResearchFee taking

Issue:

- The function reflectDevandResearchFee() do not check dev and research addresses to be excluded from reward and do not increase _tOwned balance of this addresses if needed. If them would be it will be a high issue.

```
function reflectDevandResearchFee(uint256 tDev↑, uint256 tResearch↑) private {
    uint256 currentRate =  _getRate();
    uint256 rDevelopent =  tDev↑.mul(currentRate);
    uint256 rResearch =  tResearch↑.mul(currentRate);
    _tDevelopmentTotal = _tDevelopmentTotal.add(tDev↑);
    _rOwned[devWallet] = _rOwned[devWallet].add(rDevelopent);
    _tResearchTotal = _tResearchTotal.add(tResearch↑);
    _rOwned[researchWallet] = _rOwned[researchWallet].add(rResearch);
}
```

Recommendation:

Check dev and research addresses to be excluded and increase addresses' _tOwned balance if needed.

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

- **Owner can change presale wallet.**

```solidity
function setPresaleWallet(address _presaleWallet) public onlyOwner {
    _isExcludedFromFee[_presaleWallet] = true;
    isPresaleWallet[_presaleWallet]=true;
}
```

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

```solidity
function setThreshholdForLP(uint256 threshold) external onlyOwner {
    numTokensSellToAddToLiquidity = threshold * 10**_decimals;
}
```

- **Owner can exclude from fee and rfi.**

```solidity
function excludeFromFeeAndRfi(address account) public onlyOwner {
    excludeFromFee(account);
    excludeFromRFI(account);
}
```

- **Owner can change fee rates.**

```solidity
function setRfiRatesPercents(uint8 _rfi, uint8 _lp, uint8 _research, uint8 _dev) public onlyOwner {
    feeRates.rfi = _rfi;
    feeRates.liquidity = _lp;
    feeRates.research = _research;
    feeRates.dev = _dev;
    emit FeesChanged();
}
```

- **Owner can change research and dev wallets.**

```solidity
function setWallets(address _research, address _dev) public onlyOwner {
    researchWallet = _research;
    devWallet = _dev;
    _isExcludedFromFee[_research] = true;
    _isExcludedFromFee[_dev] = true;
    emit WalletsChanged();
}
```

- **Owner can change the maximum transaction amount.**

```solidity
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner {
    uint256 _previoiusAmount = _maxTxAmount;
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(100);
    emit MaxTxAmountChanged(_previoiusAmount, _maxTxAmount);
}
```

- **Owner can exclude from the fee.**

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

Conclusion

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

Liquidity locking details are 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.