



TechRate
AUDIT COMPANY

Smart Contract Security Audit

TechRate

July, 2021

Audit Details



Audited project

Mishka Token



Deployer address

0x34D540E50a4DA79CcE7138DB8b66eA57Cdf49391



Client contacts:

Mishka Token team



Blockchain

Ethereum



Project website:

www.mishkatoken.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 Mishka Token to perform an audit of smart contracts:

<https://etherscan.io/address/0x976091738973b520a514ea206acdd008a09649de#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 12.07.2021

Contract name	Mishka Token
Contract address	0x976091738973b520A514ea206AcDD008A09649De
Total supply	1,000,000,000,000
Token ticker	MISHKA
Decimals	9
Token holders	538
Transactions count	2,593
Top 100 holders dominance	91.14%
isPublicTradingOpen	false
m_Charity	20
m_Toll initial	480
Cooldown seconds	1
Contract deployer address	0x34D540E50a4DA79CcE7138DB8b66eA57Cdf49391
Contract's current owner address	0x34d540e50a4da79cce7138db8b66ea57cdf49391

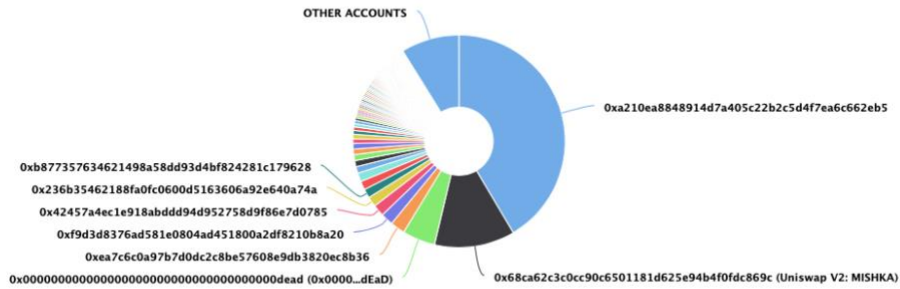
Mishka Token Token Distribution

The top 100 holders collectively own 91.14% (911,355,165,401.16 Tokens) of Mishka Token

Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 538

Mishka Token Top 100 Token Holders

Source: Etherscan.io



(A total of 911,355,165,401.16 tokens held by the top 100 accounts from the total supply of 1,000,000,000.00 token)

Mishka Token Contract Interaction Details


Time Series: Token Contract Overview

Mon 5, Jul 2021 - Sun 11, Jul 2021

Token Contract 0x976091738973b520a514ea206acdd008a09649de (Mishka Token)
Source: Etherscan.io



Mishka Token Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0xa210ea8848914d7a405c22b2c5d4f7ea6c662eb5	414,778,962,471.196814608	41.4779%
2	 Uniswap V2: MISHKA	122,450,877,427.677993734	12.2451%
3	0x0000...dEaD	50,000,000,000	5.0000%
4	0xea7c6c0a97b7d0dc2c8be57608e9db3820ec8b36	22,317,271,056.107557241	2.2317%
5	0xf9d3d8376ad581e0804ad451800a2df8210b8a20	19,000,000,000	1.9000%
6	0x42457a4ec1e918abddd94d952758d9f86e7d0785	17,639,734,017.722396419	1.7640%
7	0x236b35462188fa0fc0600d5163606a92e640a74a	15,200,000,003	1.5200%
8	0xb877357634621498a58dd93d4bf824281c179628	13,670,497,163.281480801	1.3670%
9	0xb4a6ed540e94995a9bd35e5ca6793842c186e3e4	13,247,222,226	1.3247%
10	0xcdfa2b67da46534966d3050dc95ebd2f5c5becfe	13,193,310,677.16045585	1.3193%



Contract functions details

- + Context
 - [Int] _msgSender
- + [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
- + Ownable (Context)
 - [Pub] <Constructor> #
 - [Pub] owner
 - [Pub] transferOwnership #
 - modifiers: onlyOwner
- + [Int] IUniswapV2Factory
 - [Ext] createPair #
- + [Int] IUniswapV2Router02
 - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #
 - [Ext] factory
 - [Ext] WETH
 - [Ext] addLiquidityETH (\$)
- + [Int] FTPAntiBot
 - [Ext] scanAddress #
 - [Ext] registerBlock #
- + MishkaToken (Context, IERC20, Ownable)
 - [Ext] <Fallback> (\$)
 - [Pub] <Constructor> #
 - [Pub] name
 - [Pub] symbol
 - [Pub] decimals
 - [Pub] totalSupply
 - [Pub] balanceOf
 - [Pub] transfer #
 - [Pub] allowance
 - [Pub] approve #
 - [Pub] transferFrom #
 - [Prv] _readyToSwap

- [Prv] _trader
- [Prv] _senderNotExchange
- [Prv] _txSale
- [Prv] _walletCapped
- [Prv] _isExchangeTransfer
- [Prv] _isForgiven
- [Prv] _approve #
- [Prv] _checkBot #
- [Prv] _banSeller #
- [Prv] _transfer #
- [Prv] _handleBalances #
- [Prv] _getTollBasisPoints
- [Prv] _getCharityBasisPoints
- [Prv] _payToll #
- [Prv] _swapTokensForETH #
 - modifiers: lockTheSwap
- [Prv] _disperseEth #
- [Ext] banCount
- [Ext] checkIfBanned
- [Ext] isAntiBot
- [Ext] isWhitelisted
- [Ext] isForgiven
- [Ext] isExchangeAddress
- [Ext] addLiquidity #
 - modifiers: onlyOwner
- [Ext] setTxLimit #
 - modifiers: onlyOwner
- [Ext] setTollBasisPoints #
 - modifiers: onlyOwner
- [Ext] setCharityBasisPoints #
 - modifiers: onlyOwner
- [Ext] setNumOfTokensForDisperse #
 - modifiers: onlyOwner
- [Ext] setTxLimitMax #
 - modifiers: onlyOwner
- [Pub] addBot #
 - modifiers: onlyOwner
- [Pub] aaaSendMessage #
- [Ext] aaaReadMessage
- [Pub] addBotMultiple #
 - modifiers: onlyOwner
- [Ext] removeBot #
 - modifiers: onlyOwner
- [Ext] setCoolDownSeconds #
 - modifiers: onlyOwner
- [Pub] getCoolDownSeconds
- [Ext] contractBalance
 - modifiers: onlyOwner
- [Ext] setTollAddress #
 - modifiers: onlyOwner
- [Ext] setCharityAddress #
 - modifiers: onlyOwner
- [Ext] assignAntiBot #
 - modifiers: onlyOwner
- [Ext] setAntiBotOn #

- modifiers: onlyOwner
- [Ext] setAntiBotOff #
 - modifiers: onlyOwner
- [Ext] openPublicTrading #
 - modifiers: onlyOwner
- [Ext] isPublicTradingOpen
 - modifiers: onlyOwner
- [Pub] addWhitelist #
 - modifiers: onlyOwner
- [Pub] addWhitelistMultiple #
 - modifiers: onlyOwner
- [Ext] removeWhitelist #
 - modifiers: onlyOwner
- [Ext] forgiveAddress #
 - modifiers: onlyOwner
- [Ext] rmForgivenAddress #
 - modifiers: onlyOwner
- [Ext] addExchangeAddress #
 - modifiers: onlyOwner
- [Ext] rmExchangeAddress #
 - 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 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 `addBotMultiple()` uses the loop to multiple add bot addresses from the bot list. Function will be aborted with `OUT_OF_GAS` exception if there will be a long addresses list.

```
function addBotMultiple(address[] memory _addresses↑) public onlyOwner {  
    for (uint256 i = 0; i < _addresses↑.length; i++) {  
        addBot(_addresses↑[i]);  
    }  
}
```

- The function `addWhitelistMultiple()` uses the loop to multiple add whitelist addresses from the white list. Function will be aborted with `OUT_OF_GAS` exception if there will be a long addresses list.

```
function addWhitelistMultiple(address[] memory _addresses↑) public onlyOwner {  
    for (uint256 i = 0; i < _addresses↑.length; i++) {  
        addWhitelist(_addresses↑[i]);  
    }  
}
```

Recommendation:

Check that the addresses array length is not too big.

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

- Owner can manually call addLiquidity.
- Owner can change tx limit.
- Owner can change toll fee.
- Owner can change charity fee.
- Owner can change _numOfTokensForDisperse value.
- Owner can change MaxTx to MaxWalletLimit value.
- Owner can add and remove bots.
- Owner can change cooldown seconds value.
- Owner can change toll and charity addresses.
- Owner can change antibot contract, that could not be audited.
- Owner can enable and disable antibot.
- Owner can open public trading.
- Owner can add and remove whitelist addresses.
- Owner can add and remove fee addresses.
- Owner can change m_Exchange addresses value.

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/0x68ca62c3c0cc90c6501181d625e94b4f0fdc869c>

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.