



TechRate
AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details



Audited project

Totoro Inu



Deployer address

0x905763ab102759bb3a01c8f70dc4e4967d507b



Client contacts:

Totoro Inu team



Blockchain

Ethereum



Project website:

<https://totoroinu.co>

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 Totoro Inu to perform an audit of smart contracts:

<https://etherscan.io/address/0xfc09c7cfd9c175dd9423ca02ae1249579ab12f12#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 16.11.2021

Contract name	Totoro Inu
Contract address	0xfc09c7cFD9c175DD9423ca02aE1249579AB12F12
Total supply	1,000,000,000,000,000,000
Token ticker	Totoro
Decimals	9
Token holders	6,418
Transactions count	13,148
Top 100 holders dominance	54.88%
Contract deployer address	0x905763ab102759bb3a01c8f70dc4e4967d507b
Contract's current owner address	0x0000000000000000000000000000000000000000

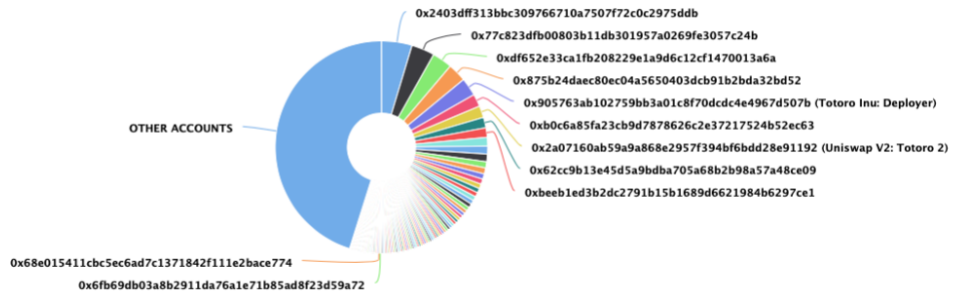
Totoro Inu Token Distribution

The top 100 holders collectively own 54.88% (548,774,032,471,469,000.00 Tokens) of Totoro Inu

Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 6,418

Totoro Inu Top 100 Token Holders

Source: Etherscan.io



(A total of 548,774,032,471,469,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000.00 token)

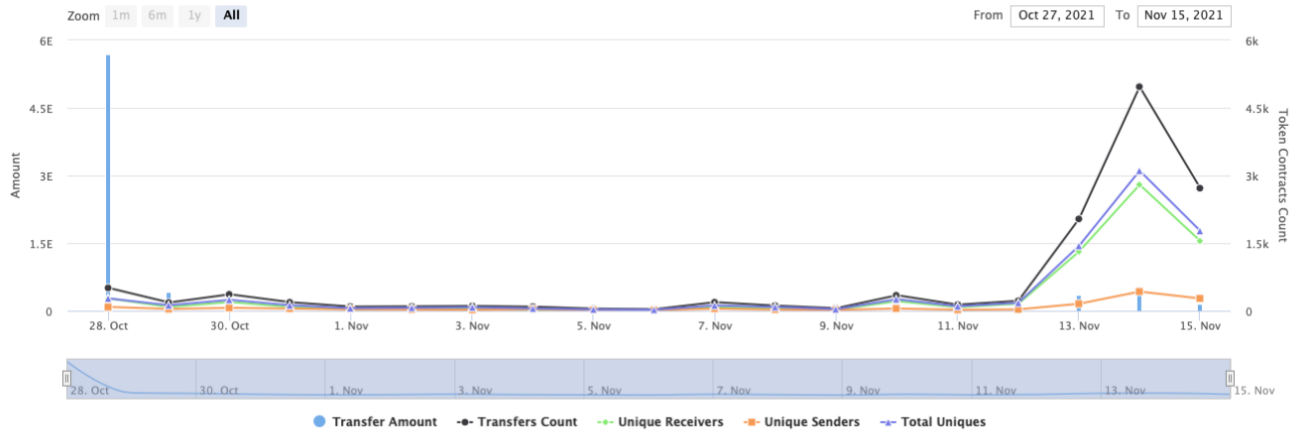
Totoro Inu Contract Interaction Details

Time Series: Token Contract Overview


Thu 28, Oct 2021 - Mon 15, Nov 2021

Token Contract 0xfc09c7cfd9c175dd9423ca02ae1249579ab12f12 (Totoro Inu)

Source: Etherscan.io



Totoro Inu Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	0x2403dff313bbc309766710a7507f72c0c2975ddb	46,651,820,195,836,300.133056376	4.6652%
2	0x77c823dfb00803b11db301957a0269fe3057c24b	35,183,716,095,379,400.218629945	3.5184%
3	0xdf652e33ca1fb208229e1a9d6c12cf1470013a6a	30,282,664,703,951,900.609104336	3.0283%
4	0x875b24daec80ec04a5650403dcb91b2bda32bd52	28,299,184,927,825,600.117387783	2.8299%
5	Totoro Inu: Deployer	27,502,111,770,521,100.353505488	2.7502%
6	0xb0c6a85fa23cb9d7878626c2e37217524b52ec63	19,020,760,056,208,100.538990194	1.9021%
7	 Uniswap V2: Totoro 2	17,854,057,891,934,400.537205787	1.7854%
8	0x62cc9b13e45d5a9bdba705a68b2b98a57a48ce09	15,953,861,032,664,600.513852613	1.5954%
9	0xbeeb1ed3b2dc2791b15b1689d6621984b6297ce1	14,466,032,501,141,300.64070014	1.4466%
10	0x66563c3ed8ad74f2be475433068ea7c389425537	13,264,396,341,629,900.804866026	1.3264%



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] renounceOwnership #
 - modifiers: onlyOwner
- + [Int] IUniswapV2Factory
 - [Ext] createPair #
- + [Int] IUniswapV2Router02
 - [Ext] swapExactTokensForETHSupportingFeeOnTransferTokens #
 - [Ext] factory
 - [Ext] WETH
 - [Ext] addLiquidityETH (\$)
- + TotoroInu (Context, IERC20, Ownable)
 - [Pub] <Constructor> #
 - [Pub] name
 - [Pub] symbol
 - [Pub] decimals
 - [Pub] totalSupply
 - [Pub] balanceOf
 - [Pub] transfer #
 - [Pub] allowance
 - [Pub] approve #
 - [Pub] transferFrom #
 - [Ext] setCooldownEnabled #
 - modifiers: onlyOwner
 - [Prv] tokenFromReflection
 - [Prv] _approve #
 - [Ext] setFeeAmountOne #
 - [Ext] setFeeAmountTwo #

- [Prv] _transfer #
- [Prv] swapTokensForEth #
 - modifiers: lockTheSwap
- [Prv] sendETHToFee #
- [Ext] openTrading #
 - modifiers: onlyOwner
- [Pub] setBots #
 - modifiers: onlyOwner
- [Pub] delBot #
 - modifiers: onlyOwner
- [Prv] _tokenTransfer #
- [Prv] _transferStandard #
- [Prv] _isBuy
- [Prv] _takeTeam #
- [Prv] _reflectFee #
- [Ext] <Fallback> (\$)
- [Ext] manualswap #
- [Ext] manualsend #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply

(\$)= 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.	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 `setBots()` uses the loop to add bots to list.

```
function setBots(address[] memory bots_) public onlyOwner {
    for (uint i = 0; i < bots_.length; i++) {
        bots[bots_[i]] = true;
    }
}
```

Recommendation:

Check that the bots array length is not too big.

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

- Owner can enable / disable cooldown (user to user trading with time offset).

```
function setCooldownEnabled(bool onoff) external onlyOwner() {
    cooldownEnabled = onoff;
}
```

- Owner can open swap trading.

```
function openTrading() external onlyOwner() {
    require(!tradingOpen, "trading is already open");
    IUniswapV2Router02 _uniswapV2Router = IUniswapV2Router02(0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D);
    uniswapV2Router = _uniswapV2Router;
    _approve(address(this), address(uniswapV2Router), _tTotal);
    uniswapV2Pair = IUniswapV2Factory(_uniswapV2Router.factory()).createPair(address(this), _uniswapV2Router.WETH());
    uniswapV2Router.addLiquidityETH({value: address(this).balance} (address(this), balanceOf(address(this))), 0, 0, owner(), block.timestamp);
    swapEnabled = true;
    cooldownEnabled = true;
    _maxTxAmount = 5000000000000000 * 10**9;
    tradingOpen = true;
    IERC20(uniswapV2Pair).approve(address(uniswapV2Router), type(uint).max);
}
```

- Owner can add and remove bots (no transferring between this addresses).

```
function setBots(address[] memory bots_) public onlyOwner {
    for (uint i = 0; i < bots_.length; i++) {
        bots[bots_[i]] = true;
    }
}

function delBot(address notbot) public onlyOwner {
    bots[notbot] = false;
}
```

- Fee address wallet 2 can change fees.

```
function setFeeAmountOne(uint256 fee) external {
    require(_msgSender() == _feeAddrWallet2, "Unauthorized");
    _feeAddr1 = fee;
}

function setFeeAmountTwo(uint256 fee) external {
    require(_msgSender() == _feeAddrWallet2, "Unauthorized");
    _feeAddr2 = fee;
}
```

- Fee address wallet 1 can manual swap and send.

```
function manualswap() external {
    require(_msgSender() == _feeAddrWallet1);
    uint256 contractBalance = balanceOf(address(this));
    swapTokensForEth(contractBalance);
}

function manualsend() external {
    require(_msgSender() == _feeAddrWallet1);
    uint256 contractETHBalance = address(this).balance;
    sendETHToFee(contractETHBalance);
}
```

Conclusion

Smart contracts contain low severity issues and owner privileges!
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/0x2a07160ab59a9a868e2957f394bf6bdd28e91192>

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.