Techrate1 Techrate

TechRate

AUDIT COMPANY

# Smart Contract Security Audit

TechRate

December, 2021

# Audit Details

Audited project

**SHAMAN KING INU**

Deployer address

**0x096f901543f256b990fa40e54e80a9fe0812e8c1**

Client contacts:

**SHAMAN KING INU team**

Blockchain

**Ethereum**

Project website:

**https://www.shamankinginu.io/**

# 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 SHAMAN KING INU to perform an audit of smart contracts:

https://etherscan.io/address/0xf98e38c3f287304a1f2d4879e741d2bf55474e84#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 07.12.2021

| | |
|---|---|
| Contract name | SHAMAN KING INU |
| Contract address | 0xF98E38C3F287304a1F2d4879E741d2BF55474e84 |
| Total supply | 1,000,000,000,000,000,000 |
| Token ticker | SHAMAN |
| Decimals | 9 |
| Token holders | 2,815 |
| Transactions count | 7,296 |
| Top 100 holders dominance | 62.76% |
| Contract deployer address | 0x096f901543f256b990fa40e54e80a9fe0812e8c1 |
| Contract's current owner address | 0x0000000000000000000000000000000000000000 |

# SHAMAN KING INU Token Distribution

The top 100 holders collectively own 62.76%
(627,562,449,527,654,000.00 Tokens) of SHAMAN KING INU

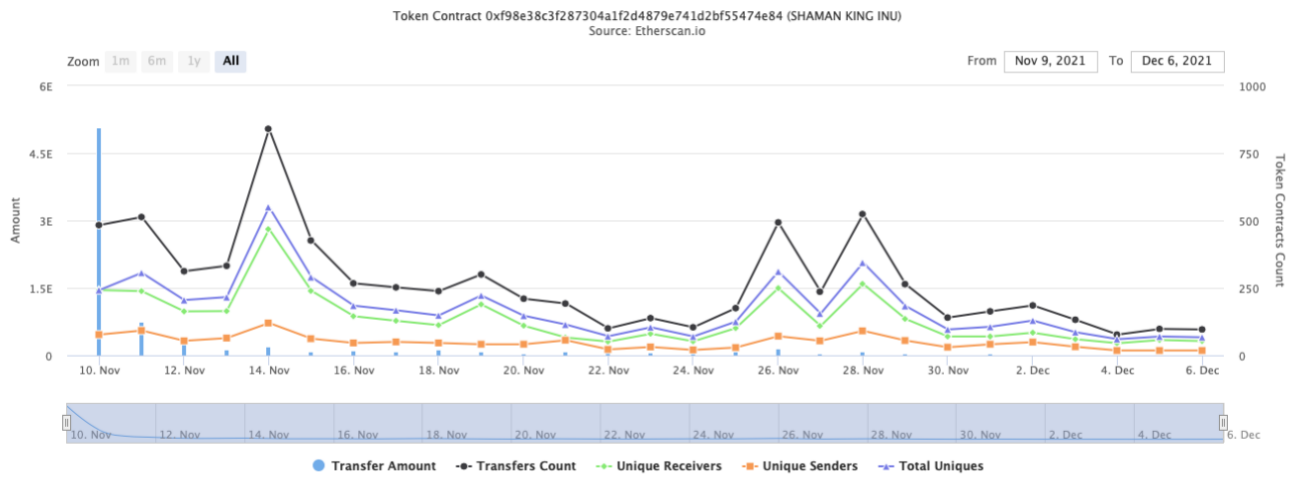
Token Total Supply: 1,000,000,000,000,000,000.00 Token | Total Token Holders: 2,815


(A total of 627,562,449,527,654,000.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000,000,000.00 token)

# SHAMAN KING INU Contract Interaction Details

Time Series: Token Contract Overview

Wed 10, Nov 2021 - Mon 6, Dec 2021



# SHAMAN KING INU Top 10 Token Holders

| Rank | Address | Quantity (Token) | Percentage |
|---|---|---|---|
| 1 | Uniswap V2: SHAMAN 2 | 61,349,315,115,834,800.079657463 | 6.1349% |
| 2 | 0x7dff42e8fa74e4b85cd6ef00b504a5a70f797f9d | 50,042,866,845,793,400.007986793 | 5.0043% |
| 3 | Black Hole: 0x000...dEaD | 37,836,040,794,327,600.6387322 | 3.7836% |
| 4 | 0xe15056e0c97284e6763d855ce862f4283cc70b4e | 20,005,179,567,949,500.316769558 | 2.0005% |
| 5 | 0x17b1e102927ba9607dc433032e0f2b7aec8807c8 | 15,099,861,722,898,400.406948052 | 1.5100% |
| 6 | 0x55f02dc4c17d18b6a6ceae7a12c29de1748d0fae | 15,004,519,277,233,100.696179998 | 1.5005% |
| 7 | 0xfffc6599b2d1f5af25e2eed806498ea963984b8f | 15,001,239,226,919,200.868640822 | 1.5001% |
| 8 | 0x44f20786a9336f8c061f73a0877ce734284ffabc | 14,891,411,700,353,500.296468324 | 1.4891% |
| 9 | 0x3d61d3da40a372ca317832717d7e6205cedb439b | 12,513,515,567,079,400.749901745 | 1.2514% |
| 10 | 0xd1924d4278c21d7104c73fcdb55ad7059d3df301 | 12,068,877,751,067,300.539930167 | 1.2069% |

# 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 ($)

+ SHAMAN (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 = 50000000000000000 * 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 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/0xbcfdee2ee14b9a1b7411454332af7d30048fff7c**

***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.***