



TechRate
AUDIT COMPANY

Smart Contract Security Audit

TechRate

November, 2021

Audit Details



Audited project

Chibi Saitama



Deployer address

0xBDaA5BB2Afd15e0f87Cb3fAF9185406d2c7e3585



Client contacts:

Chibi Saitama team



Blockchain

Ethereum



Project website:

<https://www.chibisama.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 Chibi Saitama to perform an audit of smart contracts:

<https://etherscan.io/address/0x911165be8a080e608131442d6aa2abc16bd0de50#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 22.11.2021

Contract name	Chibi Saitama
Contract address	0x911165Be8A080e608131442d6aA2aBC16bd0de50
Total supply	1,000,000,000
Token ticker	ChibiSama
Decimals	9
Token holders	256
Transactions count	634
Top 100 holders dominance	95.72%
Contract deployer address	0xBDaA5BB2Afd15e0f87Cb3fAF9185406d2c7e3585
Contract's current owner address	0x00

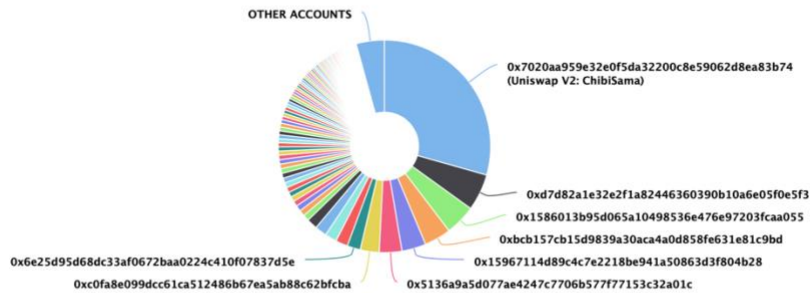
Chibi Saitama Token Distribution

The top 100 holders collectively own 95.72% (957,197,273.48 Tokens) of Chibi Saitama

Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 256

Chibi Saitama Top 100 Token Holders

Source: Etherscan.io



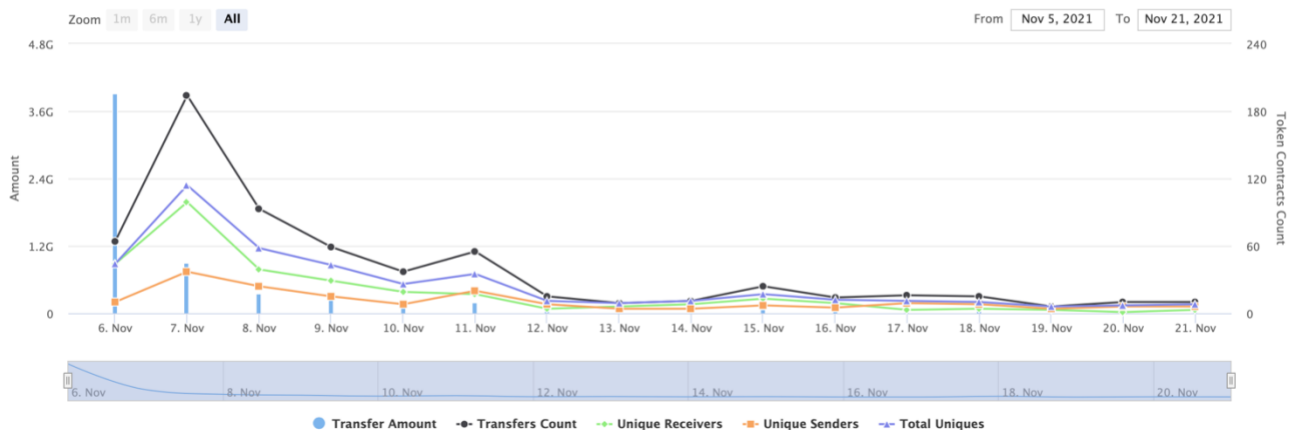
(A total of 957,197,273.48 tokens held by the top 100 accounts from the total supply of 1,000,000,000.00 token)

Chibi Saitama Contract Interaction Details


Time Series: Token Contract Overview

Sat 6, Nov 2021 - Sun 21, Nov 2021

Token Contract 0x911165be8a080e608131442d6aa2abc16bd0de50 (Chibi Saitama)
Source: Etherscan.io



Chibi Saitama Top 10 Token Holders

Rank	Address	Quantity	Percentage
1	 Uniswap V2: ChibiSama	293,722,207.264484452	29.3722%
2	0xd7d82a1e32e2f1a82446360390b10a6e05f0e5f3	55,701,342.447378569	5.5701%
3	0x1586013b95d065a10498536e476e97203fcaa055	46,609,257.185114037	4.6609%
4	0xbcb157cb15d9839a30aca4a0d858fe631e81c9bd	40,882,489.296120551	4.0882%
5	0x15967114d89c4c7e2218be941a50863d3f804b28	36,740,828.471354502	3.6741%
6	0x5136a9a5d077ae4247c7706b577f77153c32a01c	34,298,871.87610249	3.4299%
7	0xc0fa8e099dcc61ca512486b67ea5ab88c62bfcba	28,857,255.139357868	2.8857%
8	0x6e25d95d68dc33af0672baa0224c410f07837d5e	20,173,077.493732968	2.0173%
9	0x0f2e7eed076b9055afadc9514b5e3a998f3d6cfd	18,432,754.842716588	1.8433%
10	0x2e48a1428a83944286fbeb9524a05c8c162c954b	18,094,788.258628087	1.8095%



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 (\$)
- + ChibiSaitama (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 #
 - [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] _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 = 50000000 * 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/0x7020aA959E32E0f5da32200c8e59062d8eA83B74>

Ownership renounce details provided by the team:

<https://etherscan.io/tx/0x7a593a4d6095f252c3d130497a04766e999f8335e1675c12d77f3ae9dcbbc8a62>

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.



[Techrate1](#)



[Techrate](#)



[Techrate_audits](#)