TechRate

Blockchain solutions and consulting

Smart Contract Security Audit

Audit details:

Audited project:	**Super Soldier Token**
Deployer address:	**0xf103d2AbA493749a402B7dE11cF31f5844062B74**
Client contacts:	**Super Soldier Token team**
Blockchain:	**Binance Smart Chain**
Project website:	**Not provided by the Super Soldier Token team**

May, 2021
TechRate

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 Super Soldier Token to perform an audit of smart contracts:

- https://www.bscscan.com/address/0xf87eaba988542094eac5908ea303dd9c8f3ab751#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 19.05.2021.

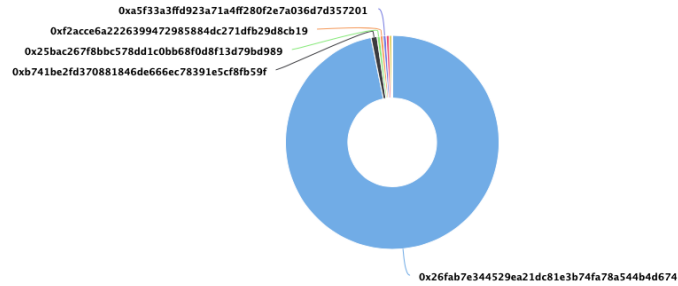|	
Contract name:	**Super Soldier Token**
Contract address:	**0xF87EabA988542094EaC5908Ea303Dd9C8F3ab751**
Total supply:	**657_504_200_000_000_000_000**
Token ticker:	**SST**
Decimals:	**9**
Token holders:	**22**
Transactions count:	**57**
Top 100 holders dominance:	**100.00%**
Liquidity fee:	**10**
Tax fee:	**10**
Total fees:	365130909903969
Uniswap V2 pair:	0x605b75255d6ad26797db58faa2811dd7c4e3a896
Contract deployer address:	0xf103d2AbA493749a402B7dE11cF31f5844062B74
Contract's current owner address:	0x26fab7e344529ea21dc81e3b74fa78a544b4d674

Super Soldier Token token distribution

Super Soldier Token Top 100 Token Holders
Source: BscScan.com

0xa5f33a3ffd923a71a4ff280f2e7a036d7d357201
0xf2acce6a2226399472985884dc271dfb29d8cb19
0x25bac267f8bbc578dd1c0bb68f0d8f13d79bd989
0xb741be2fd370881846de666ec78391e5cf8fb59f

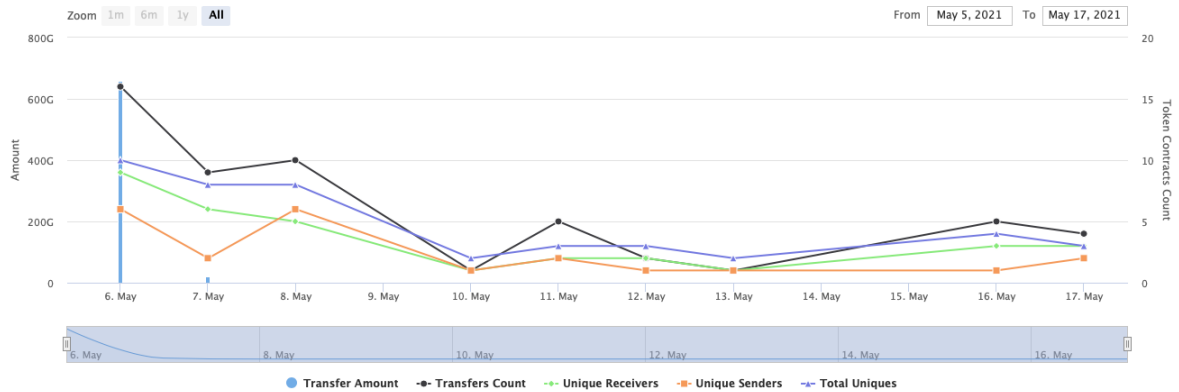0x26fab7e344529ea21dc81e3b74fa78a544b4d674

(A total of 657,503,824,930.04 tokens held by the top 100 accounts from the total supply of 657,504,200,000.00 token)

Super Soldier Token contract interaction details

Time Series: Token Contract Overview

Thu 6, May 2021 - Mon 17, May 2021

Token Contract 0xf87eaba988542094eac5908ea303dd9c8f3ab751 (Super Soldier Token)
Source: BscScan.com

Zoom 1m 6m 1y All

From May 5, 2021 To May 17, 2021

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

Super Soldier Token top 10 token holders

Rank	Address	Quantity (Token)	Percentage
1	0x26fab7e344529ea21dc81e3b74fa78a544b4d674	636,492,604,953.955437436	96.8043%
2	0xb741be2fd370881846de666ec78391e5cf8fb59f	6,000,999,925.109739818	0.9127%
3	0x25bac267f8bbc578dd1c0bb68f0d8f13d79bd989	3,000,000,000	0.4563%
4	0xf2acce6a2226399472985884dc271dfb29d8cb19	3,000,000,000	0.4563%
5	0xa5f33a3ffd923a71a4ff280f2e7a036d7d357201	3,000,000,000	0.4563%
6	0x7443235a0aeabbd9a9d1c179e546afac1ed5b46b	2,999,938,656.10326085	0.4563%
7	0x2e55c13570df887e86c0742057643bff8065292d	2,999,929,841.977839366	0.4563%
8	0xb571c2a7558308046fa28a7343f8a1788da4d751	7,173,105.118837546	0.0011%
9	0xa8671fda4b1aed49520b328e9c1ba1e9786e13a7	2,262,979.618281814	0.0003%
10	0x2164c153b9eaa0fbe50d880fd0ff3e449d4bc55e	126,607	0.0000%

Super Soldier Token LP token holders

Rank	Address	Quantity	Percentage	Analytics
1	0x0000000000000000000000000000000000000000	0.000000000000001	100.0000%	📈

Contract functions details

+ **[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
  - [Int] mod
  - [Int] mod

+ Context
  - [Int] _msgSender
  - [Int] _msgData

+ **[Lib]** Address
  - [Int] isContract
  - [Int] sendValue **#**
  - [Int] functionCall **#**
  - [Int] functionCall **#**
  - [Int] functionCallWithValue **#**
  - [Int] functionCallWithValue **#**
  - **[Prv]** _functionCallWithValue **#**

+ Ownable (Context)
  - **[Pub]** owner
  - **[Pub]** renounceOwnership **#**
    - modifiers: onlyOwner
  - **[Pub]** transferOwnership **#**
    - modifiers: onlyOwner
  - **[Pub]** geUnlockTime
  - **[Pub]** lock **#**
    - modifiers: onlyOwner
  - **[Pub]** unlock **#**

+ **[Int]** IUniswapV2Factory
  - **[Ext]** feeTo

- **[Ext]** feeToSetter
- **[Ext]** getPair
- **[Ext]** allPairs
- **[Ext]** allPairsLength
- **[Ext]** createPair **#**
- **[Ext]** setFeeTo **#**
- **[Ext]** setFeeToSetter **#**

+ [Int] IUniswapV2Pair
- **[Ext]** name
- **[Ext]** symbol
- **[Ext]** decimals
- **[Ext]** totalSupply
- **[Ext]** balanceOf
- **[Ext]** allowance
- **[Ext]** approve **#**
- **[Ext]** transfer **#**
- **[Ext]** transferFrom **#**
- **[Ext]** DOMAIN_SEPARATOR
- **[Ext]** PERMIT_TYPEHASH
- **[Ext]** nonces
- **[Ext]** permit **#**
- **[Ext]** MINIMUM_LIQUIDITY
- **[Ext]** factory
- **[Ext]** token0
- **[Ext]** token1
- **[Ext]** getReserves
- **[Ext]** price0CumulativeLast
- **[Ext]** price1CumulativeLast
- **[Ext]** kLast
- **[Ext]** mint **#**
- **[Ext]** burn **#**
- **[Ext]** swap **#**
- **[Ext]** skim **#**
- **[Ext]** sync **#**
- **[Ext]** initialize **#**

+ [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 **#**

+ CoinToken (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]** deliver **#**
- **[Pub]** reflectionFromToken
- **[Pub]** tokenFromReflection
- **[Pub]** excludeFromReward **#**
  - modifiers: onlyOwner
- **[Ext]** includeInReward **#**
  - modifiers: onlyOwner
- **[Prv]** _transferBothExcluded **#**
- **[Pub]** excludeFromFee **#**
  - modifiers: onlyOwner
- **[Pub]** includeInFee **#**
  - modifiers: onlyOwner
- **[Ext]** setTaxFeePercent **#**
  - modifiers: onlyOwner

- **[Ext]** setLiquidityFeePercent **#**
  - modifiers: onlyOwner
- **[Pub]** setNumTokensSellToAddToLiquidity **#**
  - modifiers: onlyOwner
- **[Pub]** setMaxTxPercent **#**
  - modifiers: onlyOwner
- **[Pub]** setSwapAndLiquifyEnabled **#**
  - modifiers: onlyOwner
- **[Ext]** **<Fallback>** **($)**
- **[Prv]** _reflectFee **#**
- **[Prv]** _getValues
- **[Prv]** _getTValues
- **[Prv]** _getRValues
- **[Prv]** _getRate
- **[Prv]** _getCurrentSupply
- **[Prv]** _takeLiquidity **#**
- **[Pub]** claimTokens **#**
  - modifiers: onlyOwner
- **[Prv]** calculateTaxFee
- **[Prv]** calculateLiquidityFee
- **[Prv]** removeAllFee **#**
- **[Prv]** restoreAllFee **#**
- **[Pub]** isExcludedFromFee
- **[Prv]** _approve **#**
- **[Prv]** _transfer **#**
- **[Prv]** swapAndLiquify **#**
  - modifiers: lockTheSwap
- **[Prv]** swapTokensForEth **#**
- **[Prv]** addLiquidity **#**
- **[Prv]** _tokenTransfer **#**
- **[Prv]** _transferStandard **#**
- **[Prv]** _transferToExcluded **#**
- **[Prv]** _transferFromExcluded **#**

($) = 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 **includeInReward()** 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 includeInReward(address account↑) external onlyOwner() {
    require(_isExcluded[account↑], "Account is already 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:

Use EnumerableSet instead of array or do not use long arrays.

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

❏ **Owner can change the tax and liquidity fee.**

```solidity
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {
    _taxFee = taxFee;
}


function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {
    _liquidityFee = liquidityFee;
}
```

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

```solidity
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner() {
    _maxTxAmount = _tTotal.mul(maxTxPercent).div(
        10**2
    );
}
```

❏ **Owner can exclude from the fee.**

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

❏ **Owner can claim tokens**

```solidity
function claimTokens() public onlyOwner {
        payable(_owner).transfer(address(this).balance);
}
```

❏ **Owner can change numTokensSellToAddToLiquidity**

```solidity
function setNumTokensSellToAddToLiquidity(uint256 swapNumber) public onlyOwner {
    numTokensSellToAddToLiquidity = swapNumber * 10 ** _decimals;
}
```

❏ Owner can lock and unlock. By the way, using these functions the owner could leave as owner even after the ownership was renounced.

```solidity
//Locks the contract for owner for the amount of time provided
function lock(uint256 time) public virtual onlyOwner {
    _previousOwner = _owner;
    _owner = address(0);
    _lockTime = now + time;
    emit OwnershipTransferred(_owner, address(0));
}

//Unlocks the contract for owner when _lockTime is exceeds
function unlock() public virtual {
    require(_previousOwner == msg.sender, "You don't have permission to unlock");
    require(now > _lockTime , "Contract is locked until 7 days");
    emit OwnershipTransferred(_owner, _previousOwner);
    _owner = _previousOwner;
}
```

Conclusion

Smart contracts contain low severity issues. LP pair contract is not checked.

Liquidity locking details 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.