



TechRate
AUDIT COMPANY

Smart Contract Security Audit

TechRate

June, 2021

Audit Details



Audited project

Assgard



Deployer address

0xcFAAD40056aD9092cC7a11eA4c89A49034305f75



Client contacts:

Assgard team



Blockchain

Binance Smart Chain



Project website:

<http://www.assgardtoken.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 Assgard to perform an audit of smart contracts:

<https://bscscan.com/address/0x7c52e19e4d350189bcd7b7f7a0868117e9f195e5#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 19.06.2021

Contract name	Assgard
Contract address	0x7C52e19E4d350189bcdFb7F7A0868117e9F195E5
Total supply	1,000,000,000,000,000
Token ticker	ASSG
Decimals	18
Token holders	828
Transactions count	2,073
Top 100 holders dominance	87.81%
Liquidity fee	8
Tax fee	2
Total fees	13171573739466672097875842274041
Uniswap V2 pair	0xd28558b894b9c0141f444c9a651187b39409fd03
Contract deployer address	0xcFAAD40056aD9092cC7a11eA4c89A49034305f75
Contract's current owner address	0xcfaad40056ad9092cc7a11ea4c89a49034305f75

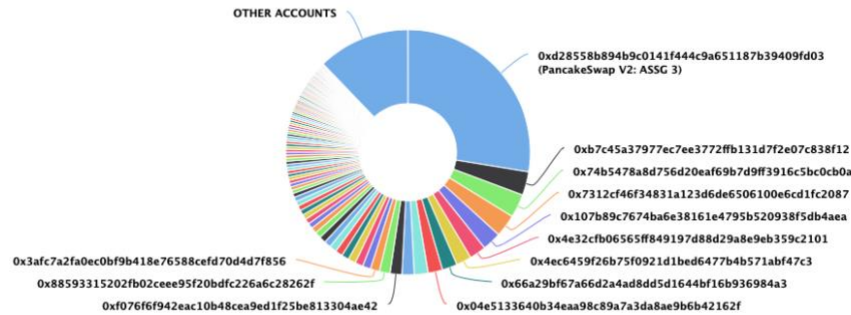
Assgard Token Distribution

The top 100 holders collectively own 87.81% (878,065,136,005,930.00 Tokens) of Assgard

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 828

Assgard Top 100 Token Holders

Source: BscScan.com



(A total of 878,065,136,005,930.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000.00 token)

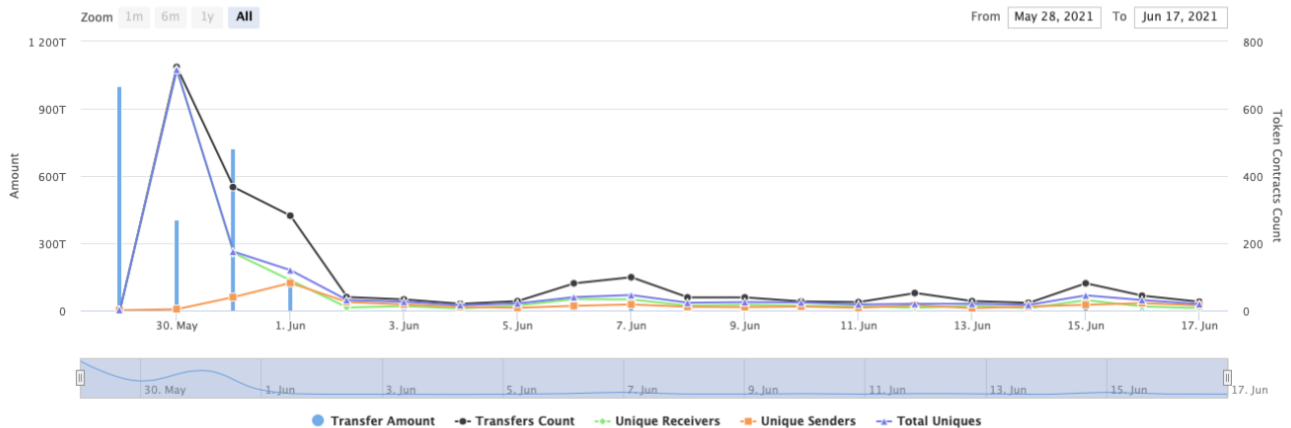
Assgard Contract Interaction Details

Time Series: Token Contract Overview


Sat 29, May 2021 - Thu 17, Jun 2021

Token Contract 0x7c52e19e4d350189bcd7f77a0868117e9f195e5 (Assgard)

Source: BscScan.com



Assgard Top 10 Token Holders

Rank	Address	Quantity (Token)	Percentage
1	 PancakeSwap V2: ASSG 3	275,834,976,348,669.398646415287880799	27.5835%
2	0xb7c45a37977ec7ee3772ffb131d7f2e07c838f12	31,044,190,414,580.980501816887324596	3.1044%
3	0x74b5478a8d756d20eaf69b7d9ff3916c5bc0cb0a	30,943,349,464,844.700443217575191418	3.0943%
4	0x7312cf46f34831a123d6de6506100e6cd1fc2087	29,579,277,450,386.831120135977608955	2.9579%
5	0x107b89c7674ba6e38161e4795b520938f5db4aea	25,108,786,630,814.123648508848286231	2.5109%
6	0x4e32cfb06565ff849197d88d29a8e9eb359c2101	21,970,969,501,786.445435390629154009	2.1971%
7	0x4ec6459f26b75f0921d1bed6477b4b571abf47c3	20,109,571,491,271.777487475649913735	2.0110%
8	0x66a29bf67a66d2a4ad8dd5d1644bf16b936984a3	19,938,955,712,297.170935348062662473	1.9939%
9	0x04e5133640b34eaa98c89a7a3da8ae9b6b42162f	19,057,802,157,471.838052932175757858	1.9058%
10	0xc63a8fd2af32d892453f943659d68459850be6bb	18,278,004,396,739.505746983810195102	1.8278%



Contract functions details

+ [Int] IUniswapV2Factory

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

+ Context

- [Int] _msgSender
- [Int] _msgData

+ [Int] IERC20

- [Ext] totalSupply
- [Ext] balanceOf
- [Ext] transfer #
- [Ext] allowance
- [Ext] approve #
- [Ext] transferFrom #

+ [Lib] Address

- [Int] isContract
- [Int] sendValue #
- [Int] functionCall #
- [Int] functionCall #
- [Int] functionCallWithValue #
- [Int] functionCallWithValue #
- [Prv] _functionCallWithValue #

+ [Lib] SafeMath

- [Int] add
- [Int] sub
- [Int] sub
- [Int] mul
- [Int] div
- [Int] div
- [Int] mod
- [Int] mod

+ Ownable (Context)

- [Int] <Constructor> #
- [Pub] owner
- [Pub] renounceOwnership #
 - modifiers: onlyOwner
- [Pub] transferOwnership #
 - modifiers: onlyOwner
- [Pub] geUnlockTime
- [Pub] lock #
 - modifiers: onlyOwner

- [Pub] unlock #
- + [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 #
- + Assgard (Context, IERC20, Ownable)
 - [Pub] <Constructor> #
 - [Pub] airDrop #
 - modifiers: onlyOwner
 - [Pub] addBotWallet #
 - modifiers: onlyOwner
 - [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
 - [Ext] setLiquidityFeePercent #
 - modifiers: onlyOwner
 - [Ext] setMaxTxPercent #
 - modifiers: onlyOwner
 - [Pub] setSwapAndLiquifyEnabled #
 - modifiers: onlyOwner
 - [Ext] <Fallback> (\$)
 - [Prv] _reflectFee #
 - [Prv] _getValues
 - [Prv] _tokenTransfer #
 - [Ext] setTaxFeePercent #
 - modifiers: onlyOwner
 - [Pub] includeInFee #
 - modifiers: onlyOwner
 - [Prv] _getTValues
 - [Prv] _getRValues
 - [Prv] _getRate
 - [Prv] _getCurrentSupply
 - [Prv] _takeLiquidity #
 - [Prv] calculateTaxFee
 - [Prv] calculateLiquidityFee
 - [Prv] removeAllFee #

- [Prv] restoreAllFee #
- [Pub] isExcludedFromFee
- [Prv] swapTokensForEth #
- [Prv] _transferToExcluded #
- [Prv] _transferFromExcluded #
- [Prv] _approve #
- [Prv] addLiquidity #
- [Prv] _transferStandard #
- [Prv] _transfer #
- [Prv] swapAndLiquify #
 - modifiers: lockTheSwap

(\$) = 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.

```
function includeInReward(address account) external onlyOwner() {
    require(!_excluded[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.

```
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:

Check that the excluded array length is not too big.

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

- Owner can change the tax and liquidity fee.

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

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

- Owner can change the maximum transaction amount.

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

- Owner can exclude from the fee.

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

- Owner can run airdrop.

```
function airDrop(address recipient, uint256 value) public onlyOwner returns(bool){
    removeAllFee();
    _transfer(_msgSender(), recipient, value * 10 ** 18);
    restoreAllFee();
    return true;
}
```

- Owner can add but can't remove bot wallet.

```
function addBotWallet(address wallet) public onlyOwner returns(bool){
    _botWallets[wallet] = true;
    return true;
}
```


- Owner can lock and unlock. By the way, using these functions the owner could retake privileges even after the ownership was renounced.

```
//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! Liquidity pair contract's security is not checked due to out of scope.

Liquidity locking details provided by the team:

<https://bscscan.com/token/0xd28558B894b9c0141f444C9a651187b39409Fd03?a=0xa36037dc26c5c02e864eba969a312320e6487269>

<https://bscscan.com/tx/0x894f71941715a7324b663ab9c4156bad0d62d9875ae75975f8cbc4d6773c5f52>

<div><div> BSCrypt Lock</div><div>Home Audit Explore About Lockup</div><div>0xcFAAD40056aD9092cC7a11eA4c89A49034305f75</div></div>				
Your locks				
Token Name	Token Address	Deposit	Remaining Days	Withdrawn
Pancake LPs	0xd285588894b9c0141f444C9a651187b39409Fd03	11889710.6900000001341104507	129 days, 23 hours, 46 minutes and 35 seconds	false
Pancake LPs	0xd285588894b9c0141f444C9a651187b39409Fd03	11889706.000000000000000000	99 days, 23 hours, 45 minutes and 48 seconds	false
Pancake LPs	0xd285588894b9c0141f444C9a651187b39409Fd03	11889706.000000000000000000	69 days, 23 hours, 44 minutes and 51 seconds	false
Pancake LPs	0xd285588894b9c0141f444C9a651187b39409Fd03	11889706.000000000000000000	39 days, 23 hours, 44 minutes and 3 seconds	false
Pancake LPs	0xd285588894b9c0141f444C9a651187b39409Fd03	11889706.000000000000000000	9 days, 23 hours, 42 minutes and 32 seconds	false

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](#)