



Smart Contract Security Audit

Audit details:

Audited project:	TheChadToken
Deployer address:	0xe7b167fc970c0866f616184b67aba44ae76094db
Client contacts:	TheChadToken team
Blockchain:	Binance Smart Chain
Project website:	Not provided

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 TheChadToken to perform an audit of smart contracts:

- <https://bscscan.com/address/0xea8eacce22bbb89709482c0100e75e7ab90f53f4#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 10.05.2021.

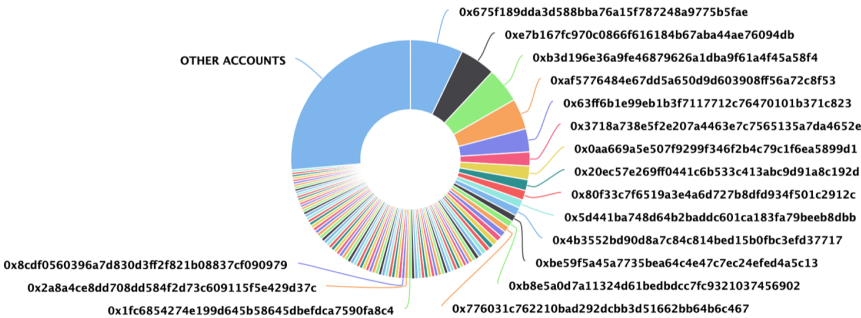
Contract name:	TheChadToken
Contract address:	0xea8eacce22bbb89709482c0100e75e7ab90f53f4
Total supply:	1000000000000000000000000
Token ticker:	CHAD
Decimals:	9
Token holders:	2993
Transactions count:	4573
Top 100 holders dominance:	73.58 %
Liquidity fee:	5
Tax fee:	5
Total fees:	0
Uniswap V2 pair:	0x675f189dda3d588bba76a15f787248a9775b5fae
Contract deployer address:	0xe7b167fc970c0866f616184b67aba44ae76094db
Contract's current owner address:	0xe7b167fc970c0866f616184b67aba44ae76094db

TheChadToken token distribution

The top 100 holders collectively own 73.58% (735,827,740,393,505.00 Tokens) of TheChadToken | Token Total Supply: 1,000,000,000,000,000.00 Token | Total Token Holders: 2,993


TheChadToken Top 100 Token Holders

Source: BscScan.com



(A total of 735,827,740,393,505.00 tokens held by the top 100 accounts from the total supply of 1,000,000,000,000.00 token)

TheChadToken top 10 holders

Rank	Address	Quantity (Token)	Percentage
1	 0x675f189dda3d588bba76a15f787248a9775b5fae	71,475,952,578,723.963641206	7.1476%
2	0xe7b167fc970c0866f616184b67aba44ae76094db	48,221,989,230,028.591567773	4.8222%
3	0xb3d196e36a9fe46879626a1dba9f61a4f45a58f4	47,318,153,887,184.869599882	4.7318%
4	0xaf5776484e67dd5a650d9d603908ff56a72c8f53	41,711,366,435,995.161555608	4.1711%
5	0x63ff6b1e99eb1b3f7117712c76470101b371c823	31,435,056,039,822.988600754	3.1435%
6	0x3718a738e5f2e207a4463e7c7565135a7da4652e	18,963,168,579,346.000686575	1.8963%
7	0x0aa669a5e507f9299f346f2b4c79c1f6ea5899d1	18,527,710,656,188.264758386	1.8528%
8	0x20ec57e269ff0441c6b533c413abc9d91a8c192d	14,837,246,167,283.159927689	1.4837%
9	0x80f33c7f6519a3e4a6d727b8dfd934f501c2912c	13,040,922,947,785.442671667	1.3041%
10	0x5d441ba748d64b2baddc601ca183fa79beeb8dbb	11,426,952,370,108.46492308	1.1427%

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)

- [Int] <Constructor> #
- [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 #

+ TheChadToken (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] updateRouterAndPair #
 - modifiers: onlyOwner
- [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
- [Ext] setMaxTxPercent #
 - modifiers: onlyOwner
- [Pub] setSwapAndLiquifyEnabled #
 - modifiers: onlyOwner
- [Ext] <Fallback> (\$)
- [Prv] _reflectFee #
- [Prv] _getValues
- [Prv] _getTValues
- [Prv] _getRValues
- [Prv] _getRate
- [Prv] _getCurrentSupply
- [Prv] _takeLiquidity #
- [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 #
- [Pub] safeTransferETH #
 - modifiers: onlyOwner
- [Pub] safeTransfer #
 - modifiers: onlyOwner

(\$) = 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(!_isExcluded[account↑], "Account is already excluded");
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_excluded[i] == account↑) {
            // updating _rOwned to make sure the balances stay the same
            if (_tOwned[account↑] > 0)
            {
                uint256 newrOwned = _tOwned[account↑].mul(_getRate());
                _rTotal = _rTotal.sub(_rOwned[account↑]-newrOwned);
                _tFeeTotal = _tFeeTotal.add(_rOwned[account↑]-newrOwned);
                _rOwned[account↑] = newrOwned;
            }
            else
            {
                _rOwned[account↑] = 0;
            }

            _tOwned[account↑] = 0;
            _excluded[i] = _excluded[_excluded.length - 1];
            _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:

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

2. Wrong total fees display

Issue:

There will be wrong total fees displayed, because of adding the value with the wrong type to `_tFeeTotal`.

```

// updating _rOwned to make sure the balances stay the same
if (_tOwned[account↑] > 0)
{
    uint256 newrOwned = _tOwned[account↑].mul(_getRate());
    _rTotal = _rTotal.sub(_rOwned[account↑]-newrOwned);
    _tFeeTotal = _tFeeTotal.add(_rOwned[account↑]-newrOwned);
    _rOwned[account↑] = newrOwned;
}

```

Recommendation:

You should divide the `_rOwner[account]-newrOwner` by the rate before adding this to the total fees.

Owner privileges

- ☐ Owner can change the tax and liquidity fee.
- ☐ Owner can change the maximum transaction amount.
- ☐ Owner can exclude from the fee.
- ☐ Owner can change the router and pair contracts.

```

function updateRouterAndPair(address _uniswapV2Router↑, address _uniswapV2Pair↑) public onlyOwner() {
    _uniswapV2Router = IUniswapV2Router02(_uniswapV2Router↑);
    _uniswapV2Pair = _uniswapV2Pair↑;
}

```

- ☐ Owner can withdraw the funds from this contract.

```
function safeTransferETH(address to↑, uint value↑) public onlyOwner {
    (bool success,) = to↑.call{value:value↑}(new bytes(0));
    require(success, 'TransferHelper: ETH_TRANSFER_FAILED');
}
```

- ❑ Owner can withdraw the tokens from this contract.

```
function safeTransfer(address token↑, address to↑, uint value↑) public onlyOwner {
    // bytes4(keccak256(bytes('transfer(address,uint256)')));
    (bool success, bytes memory data) = token↑.call(abi.encodeWithSelector(0xa9059cbb, to↑, value↑));
    require(success && (data.length == 0 || abi.decode(data, (bool))), 'TransferHelper: TRANSFER_FAILED');
}
```

Conclusion

Smart contracts do not contain high severity issues! Liquidity pair contract's security is not checked due to out of scope.

Liquidity locking details provided by the team -

<https://dxsale.app/app/pages/dxlockview?id=0&add=0xe7b167FC970c0866f616184B67aBa44aE76094dB&type=lplock&chain=BSC>

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.