



# Smart Contract Security Audit

## Audit details:

|                   |  |
|-------------------|--|
| Audited project:  | Crucifearous Finance                       |
| Deployer address: | 0x357b174d3690998845c0a5d3b2762e8c600bb814 |
| Client contacts:  | Crucifearous Finance team                  |
| Blockchain:       | Binance Smart Chain                        |
| Project website:  | Not provided                               |

# 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 Crucifearous Finance to perform an audit of smart contracts:

- <https://bscscan.com/address/0x6bf3a93793e52f75543c92afec4636559988e3c1#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 13.05.2021.

|                                   |  |
|-----------------------------------|--|
| Contract name:                    | Crucifearous Finance                       |
| Contract address:                 | 0x6bf3a93793e52f75543c92afec4636559988e3c1 |
| Total supply:                     | 12845742213087519726                       |
| Token ticker:                     | CIFI                                       |
| Decimals:                         | 9  |
| Token holders:                    | 939  |
| Transactions count:               | 4912                                       |
| Top 100 holders dominance:        | 97.26 %                                    |
| Burn fee:                         | 400  |
| Tax fee:                          | 300  |
| Charity fee:                      | 300  |
| Total fees:                       | 12845742213087519726                       |
| Contract deployer address:        | 0x357b174d3690998845c0a5d3b2762e8c600bb814 |
| Contract's current owner address: | 0x81e197831912506dc40117c895723f3b9ae96195 |

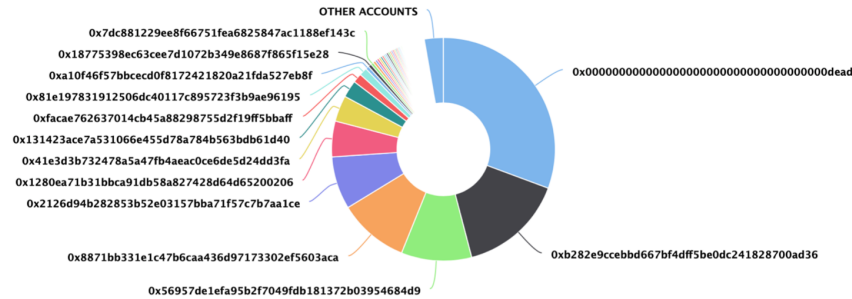
## Crucifearous Finance token distribution

💡 The top 100 holders collectively own 97.26% (955,916,198,153.43 Tokens) of Crucifearous Finance

💡 Token Total Supply: 982,872,905,422.94 Token | Total Token Holders: 939

### Crucifearous Finance Top 100 Token Holders

Source: BscScan.com



(A total of 955,916,198,153.43 tokens held by the top 100 accounts from the total supply of 982,872,905,422.94 token)

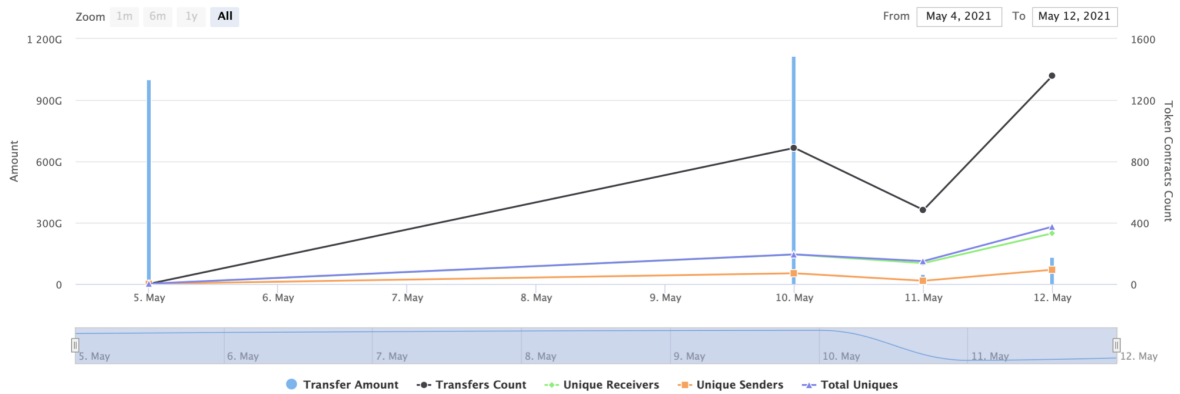
## Crucifearous Finance contract interaction details

### Time Series: Token Contract Overview


Wed 5, May 2021 - Wed 12, May 2021

Token Contract 0x6bf3a93793e52f75543c92afec4636559988e3c1 (Crucifearous Finance)

Source: BscScan.com



# Crucifearous Finance top 10 token holders

| Rank | Address  | Quantity (Token)          | Percentage |
|------|--|---------------------------|------------|
| 1    | <a href="#">0x0000000000000000000000000000000000dead</a>   | 301,463,034,301.197042993 | 30.6716%   |
| 2    | <a href="#">0xb282e9ccebbd667bf4dff5be0dc241828700ad36</a>   | 150,000,000,000           | 15.2614%   |
| 3    | <a href="#">0x56957de1efa95b2f7049fdb181372b03954684d9</a>   | 100,000,000,000           | 10.1743%   |
| 4    | <a href="#">0x8871bb331e1c47b6caa436d97173302ef5603aca</a>   | 100,000,000,000           | 10.1743%   |
| 5    | <a href="#">0x2126d94b282853b52e03157bba71f57c7b7aa1ce</a>   | 74,908,317,464.576704285  | 7.6214%    |
| 6    | <a href="#">0x1280ea71b31bbca91db58a827428d64d65200206</a>   | 50,874,179,980.489804629  | 5.1761%    |
| 7    | <a href="#">0x41e3d3b732478a5a47fb4aeac0ce6de5d24dd3fa</a>   | 37,137,461,756.481360586  | 3.7785%    |
| 8    |  <a href="#">0x131423ace7a531066e455d78a784b563bdb61d40</a> | 24,383,479,229.915912531  | 2.4808%    |
| 9    | <a href="#">0xfacae762637014cb45a88298755d2f19ff5bbaff</a>   | 13,162,152,221.564032244  | 1.3392%    |
| 10   | <a href="#">0x81e197831912506dc40117c895723f3b9ae96195</a>   | 11,239,693,057.438746964  | 1.1436%    |

# Contract functions details

## + Context

- [Int] \_msgSender
- [Int] \_msgData

## + [Int] IBEP20

- [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

## + [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

## + CoinToken (Context, IBEP20, 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] isExcluded
- [Pub] isCharity
- [Pub] totalFees
- [Pub] totalBurn
- [Pub] totalCharity
- [Pub] deliver #
- [Pub] reflectionFromToken
- [Pub] tokenFromReflection
- [Ext] excludeAccount #
  - modifiers: onlyOwner
- [Ext] includeAccount #
  - modifiers: onlyOwner
- [Ext] setAsCharityAccount #
  - modifiers: onlyOwner
- [Pub] burn #
- [Pub] updateFee #
  - modifiers: onlyOwner
- [Int] \_burn #
- [Pub] mint #
  - modifiers: onlyOwner
- [Prv] \_approve #
- [Prv] \_transfer #
- [Prv] \_transferStandard #
- [Prv] \_standardTransferContent #
- [Prv] \_transferToExcluded #
- [Prv] \_excludedFromTransferContent #
- [Prv] \_transferFromExcluded #
- [Prv] \_excludedToTransferContent #
- [Prv] \_transferBothExcluded #
- [Prv] \_bothTransferContent #
- [Prv] \_reflectFee #
- [Prv] \_getValues
- [Prv] \_getTBasics
- [Prv] getTTransferAmount
- [Prv] \_getRBasics
- [Prv] \_getRTransferAmount
- [Prv] \_getRate
- [Prv] \_getCurrentSupply
- [Prv] \_sendToCharity #
- [Prv] removeAllFee #



- [Prv] restoreAllFee #
- [Prv] \_getTaxFee

(\$) = payable function

# = non-constant function

# Issues Checking Status

| №  | Issue description.   | Checking status |
|----|--|-----------------|
| 1  | Compiler errors.   | Passed          |
| 2  | Race conditions and Reentrancy.<br>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.  | High issues     |
| 19 | Cross-function race conditions.                                    | Passed          |
| 20 | Safe Open Zeppelin contracts<br>implementation and usage.          | Passed          |
| 21 | Fallback function security.  | Passed          |

# Security Issues

## High Severity Issues

### 1. Wrong burn and mint

Issue:

In burn and mint functions there are wrong values adding because of not converting `_value`. `_rOwned` and `_tTotal` show balances in different modes and same values will be added / subtracted to them, which will make it wrong.

```
function _burn(address _who↑, uint256 _value↑) internal {
    require(_value↑ <= rOwned[_who↑]);
    rOwned[_who↑] = rOwned[_who↑].sub(_value↑);
    tTotal = tTotal.sub(_value↑);
    emit Transfer(_who↑, address(0), _value↑);
}

function mint(address account↑, uint256 amount↑) onlyOwner() public {
    tTotal = tTotal.add(amount↑);
    rOwned[account↑] = rOwned[account↑].add(amount↑);
    emit Transfer(address(0), account↑, amount↑);
}
```

Recommendation:

Please check if the addresses are included in reward or not and add the values correctly by multiplying by the rate.

## 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↑) {
            _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:

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

## Conclusion

Smart contracts contain high severity issues!

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.*