



## SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



**InfinityRIZE**  
\$RIZE



**03/01/2022**



# TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 WEBSITE DIAGNOSTIC
- 5-6 AUDIT OVERVIEW
- 7-8 OWNER PRIVILEGES
- 9 CONCLUSION AND ANALYSIS
- 10 TOKEN DETAILS
- 11 INFINITYRIZE TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS
- 12 TECHNICAL DISCLAIMER



# DISCLAIMER

The information provided on this analysis document is only for general information and should not be used as a reason to invest.

FreshCoins Team will take no payment for manipulating the results of this audit.

The score and the result will stay on this project page information on our website <https://freshcoins.io>

FreshCoins Team does not guarantees that a project will not sell off team supply, or any other scam strategy ( RUG or Honeypot etc )



# INTRODUCTION

FreshCoins (Consultant) was contracted by InfinityRIZE (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x05D29BE85f91975fd786e3fd6587dB7f78A44e67

Network: Binance Smart Chain (BSC)

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 03/01/2022



# WEBSITE DIAGNOSTIC

<https://www.infinityrize.com/>



0-49



50-89



90-100



Performance



Accessability



Best Practices



SEO



Progressive  
Web App

## Metrics



First Contentful Paint

1.2 s



Time to interactive

6.4 s



Speed Index

4.2 s



Total Blocking Time

120 ms



Large Contentful Paint

4.1 s



Cumulative Layout Shift

0.001

## Issues found

---

Reduce unused JavaScript

---

Enable text compression

---

Minify JavaScript

---

Avoid enormous network payloads - Total size was 9,392 KiB

---

Ensure text remains visible during webfont load

---

Minify JavaScript

---

Ensure text remains visible during webfont load

---

Image elements do not have explicit width and height

---

# AUDIT OVERVIEW



**Security Score**



**Static Scan**  
Automatic scanning for common vulnerabilities



**ERC Scan**  
Automatic checks for ERC's conformance

0 **High**

0 **Medium**

0 **Low**

0 **Optimizations**

0 **Informational**



No.	Issue description	Checking Status
1	Compiler Errors / Warnings	Passed
2	Reentrancy and Cross-function	Passed
3	Front running	Passed
4	Timestamp dependence	Passed
5	Integer Overflow and Underflow	Passed
6	Reverted DoS	Passed
7	DoS with block gas limit	Passed
8	Methods execution permissions	Passed
9	Exchange rate impact	Passed
10	Malicious Event	Passed
11	Scoping and Declarations	Passed
12	Uninitialized storage pointers	Passed
13	Design Logic	Passed
14	Safe Zeppelin module	Passed

# OWNER PRIVILEGES

## Contract owner can exclude wallet from fee

```
function setExcludeFromFee(address account, bool value) external onlyOwner {
    require(_isExcludedFromFee[account] != value, "Already set");
    _isExcludedFromFee[account] = value;
}
```

## Contract owner can exclude/include wallet from reward

```
function excludeFromReward(address account) public onlyOwner {
    require(!_isExcludedFromReward[account], "Account is already excluded");
    if (_reflectionBalance[account] > 0) {
        _tokenBalance[account] = tokenFromReflection(
            _reflectionBalance[account]
        );
    }
    _isExcludedFromReward[account] = true;
    _excludedFromReward.push(account);
}

function includeInReward(address account) external onlyOwner {
    require(_isExcludedFromReward[account], "Account is already included");
    for (uint256 i = 0; i < _excludedFromReward.length; i++) {
        if (_excludedFromReward[i] == account) {
            _excludedFromReward[i] = _excludedFromReward[_excludedFromReward.length - 1];
            _tokenBalance[account] = 0;
            _isExcludedFromReward[account] = false;
            _excludedFromReward.pop();
            break;
        }
    }
}
```

## Contract owner can activate/deactivate taxes

```
function setFeeActive(bool value) external onlyOwner{
    require(isFeeActive != value, "Already set");
    isFeeActive = value;
}
```

## Contract owner can set max tx amount

```
function setMaxTxAmount(uint256 _txNumerator, uint256 _divisor) external onlyOwner() {
    require(_txNumerator > 1 && _txNumerator <= 50 && _divisor > 600 && _divisor <= 1000, "Invalid MaxTX-Amount. Numerator >1 & <=50. Divisor >600 & <=1000");
    maxTxAmount = (_tokenTotal * _txNumerator) / _divisor;
}
```

## Contract owner can change the fees

```
function setTransferFee(uint256 buy, uint256 sell, uint256 p2p) external onlyOwner {
    require(buy <= 2500 && sell <= 2500 && p2p <= 2500, "Invalid fee");
    _transferFee[0] = buy;
    _transferFee[1] = sell;
    _transferFee[2] = p2p;
}

function setRewardsFee(uint256 buy, uint256 sell, uint256 p2p) external onlyOwner {
    require(buy <= 2500 && sell <= 2500 && p2p <= 2500, "Invalid fee");
    _rewardsFee[0] = buy;
    _rewardsFee[1] = sell;
    _rewardsFee[2] = p2p;
}

function setInfinityFee(uint256 buy, uint256 sell, uint256 p2p) external onlyOwner {
    require(buy <= 2500 && sell <= 2500 && p2p <= 2500, "Invalid fee");
    _infinityFee[0] = buy;
    _infinityFee[1] = sell;
    _infinityFee[2] = p2p;
}
```

## Contract owner can change swap settings

```
function setSwapEnabled(bool enabled) external onlyOwner {
    require(swapEnabled != enabled, "Already set");
    swapEnabled = enabled;
}
```



# CONCLUSION AND ANALYSIS



Smart Contracts within the scope were manually reviewed and analyzed with static tools.



Audit report overview contains all found security vulnerabilities and other issues in the reviewed code.



Found no issue during the first review.

# TOKEN DETAILS

## Details

Buy fees: N/A

Sell fees: N/A

Max TX: N/A

Max Sell: N/A

## Honeypot Risk

Ownership: Owned

Blacklist: Not detected

Modify Max TX: Detected

Modify Max Sell: Not detected

Disable Trading: Not detected

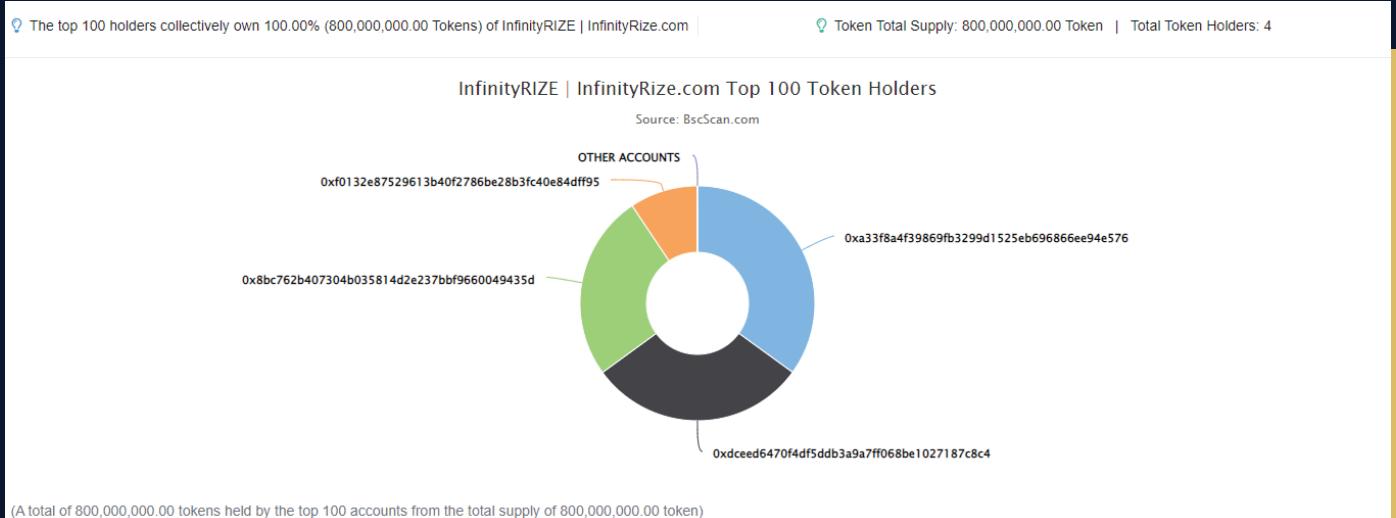
## Rug Pull Risk

Liquidity: N/A

Holders: Clean



# INFINITYRIZE TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS



Rank	Address	Quantity (Token)	Percentage
1	<a href="#">0xa33f8a4f39869fb3299d1525eb696866ee94e576</a>	280,000,000	35.0000%
2	<a href="#">0xdceed6470f4df5ddb3a9a7ff068be1027187c8c4</a>	240,000,000	30.0000%
3	<a href="#">0x8bc762b407304b035814d2e237bbf9660049435d</a>	205,000,000	25.6250%
4	<a href="#">0xf0132e87529613b40f2786be28b3fc40e84dff95</a>	75,000,000	9.3750%

# TECHNICAL DISCLAIMER

Smart contracts are deployed and executed on the blockchain platform. The platform, its programming language, and other software related to the smart contract can have its vulnerabilities that can lead to hacks. The audit can't guarantee the explicit security of the audited project / smart contract.

