



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



TreeToken
\$TREE

29/04/2022

TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-6 OWNER PRIVILEGES
- 7 CONCLUSION AND ANALYSIS
- 8 TOKEN DETAILS
- 9 TREETOKEN ANALYTICS & TOP 10 TOKEN HOLDERS
- 10 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 TreeToken (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x4ea13b58EF2eAD91bF969F2815e0766d1ae7866F

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 29/04/2022



AUDIT OVERVIEW



Security Score



Static Scan
Automatic scanning for common vulnerabilities



ERC Scan
Automatic checks for ERC's conformance



High



Medium



Low



Optimizations



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't mint tokens after initial contract deploy

Contract owner can't exclude an address from transactions

Contract owner can exclude/include wallet from tax

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

function includeInFee(address account) public onlyOwner {
    _isExcludedFromFee[account] = false;
}
```

Contract owner can exclude/include wallet from rewards

```
function excludeFromReward(address account) public onlyOwner() {
    require(!_isExcluded[account], "Account is already excluded");
    if(_rOwned[account] > 0) {
        _tOwned[account] = tokenFromReflection(_rOwned[account]);
    }
    _isExcluded[account] = true;
    _excluded.push(account);
}

function includeInReward(address account) external onlyOwner() {
    require(_isExcluded[account], "Account is already included");
    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;
        }
    }
}
```

Contract owner can change `_devWalletAddress` address

Current value:

`_devWalletAddress: 0xc8368b81147fcf8c918046973deb57a25be0b45c`

```
function setDevWalletAddress(address _addr) public onlyOwner {
    _devWalletAddress = _addr;
}
```

Contract owner can change fees up to 11%

```
function setTaxFeePercent(uint256 taxFee) external onlyOwner() {  
    require(taxFee <= 2, "Tax fee is over 10%");  
    _taxFee = taxFee;  
}  
  
function setDevFeePercent(uint256 devFee) external onlyOwner() {  
    require(devFee <= 6, "Dev fee is over 10%");  
    _devFee = devFee;  
}  
  
function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner() {  
    require(liquidityFee <= 3, "Liquidity fee is over 10%");  
    _liquidityFee = liquidityFee;  
}
```

Contract owner can change max tx limitations

```
function setMaxTxPercent(uint256 maxTxPercent) public onlyOwner {  
    _maxTxAmount = maxTxPercent * 10 ** _decimals;  
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {  
    emit OwnershipTransferred(_owner, address(0));  
    _owner = address(0);  
}
```

Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {  
    require(newOwner != address(0), 'Ownable: new owner is the zero address');  
    emit OwnershipTransferred(_owner, newOwner);  
    _owner = newOwner;  
}
```



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:	11%
Sell fees:	11%
Max TX:	500,000,000,000
Max Sell:	N/A

Honeypot Risk

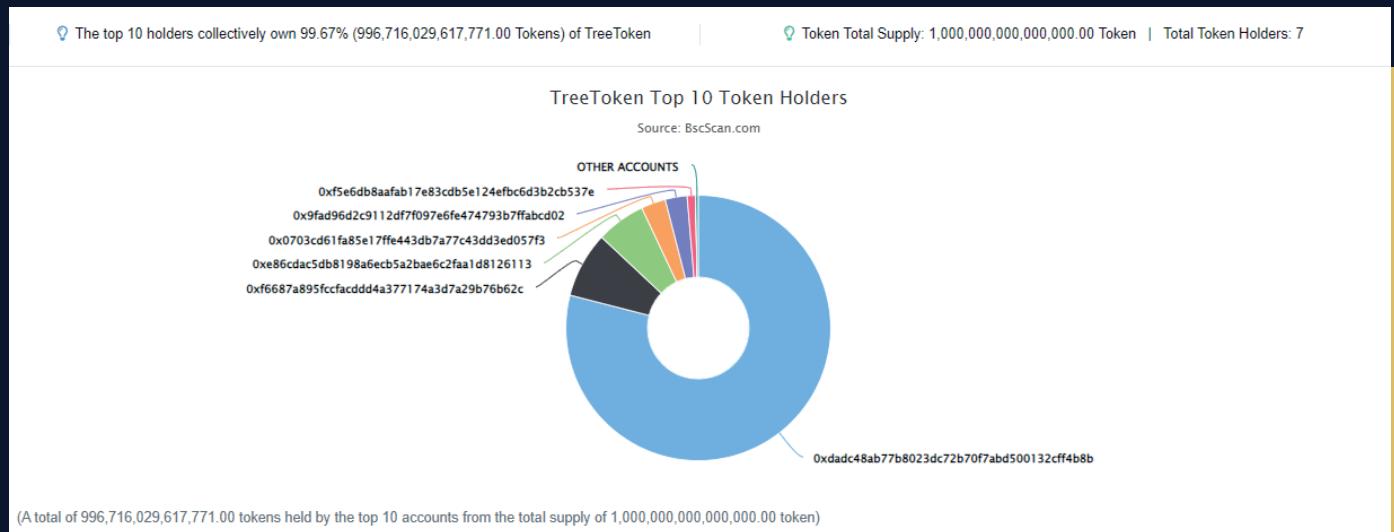
Ownership:	Owned
Blacklist:	Not detected
Modify Max TX:	Detected
Modify Max Sell:	Not detected
Disable Trading:	Not detected

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean



TREETOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



Rank	Address	Quantity (Token)	Percentage
1	0xdadc48ab77b8023dc72b70f7abd500132cff4b8b	789,888,000,000,000	78.9888%
2	0xf6687a895fccfacdd4a377174a3d7a29b76b62c	80,000,000,000,000	8.0000%
3	0xe86cdac5db8198a6ecb5a2bae6c2faa1d8126113	60,000,000,000,000	6.0000%
4	0x0703cd61fa85e17ffe443db7a77c43dd3ed057f3	30,000,000,000,000	3.0000%
5	0x9fad96d2c9112df7f097e6fe474793b7ffabcd02	26,716,029,617,770,662397438	2.6716%
6	0xf5e6db8aafab17e83cdb5e124efbc6d3b2cb537e	10,000,000,000,000	1.0000%
7	0x13f2c0e3885352cdab38c2c01cdb723b14b289be	112,000,000,000	0.0112%

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.

