



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Dragon Inu
\$DIN

05/03/2022

TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-6 OWNER PRIVILEGES
- 7 CONCLUSION AND ANALYSIS
- 8 TOKEN DETAILS
- 9 DRAGON INU TOKEN 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 Dragon Inu (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x362d5aeA66b67c96ba51931D6777993C8AA69254

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 05/03/2022



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 exclude an address from transactions.

Contract owner can't mint tokens after initial contract deploy

Contract owner can exclude/include wallet(s) from tax

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    require(_isExcludedFromFees[account] != excluded, "DIN: Account is already the value of 'excluded'");
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}

function excludeMultipleAccountsFromFees(address[] calldata accounts, bool excluded) public onlyOwner {
    for(uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFees[accounts[i]] = excluded;
    }

    emit ExcludeMultipleAccountsFromFees(accounts, excluded);
}
```

Contract owner can exclude/include wallet from tx limitations & tax

```
function setExcludeFromMaxTx(address _address, bool value) public onlyOwner {
    _isExcludedFromMaxTx[_address] = value;
}

function setExcludeFromAll(address _address) public onlyOwner {
    _isExcludedFromMaxTx[_address] = true;
    _isExcludedFromFees[_address] = true;
}
```

Contract owner can change swap settings

```
function SetSwapTokensAtAmount(uint256 newLimit) external onlyOwner {
    swapTokensAtAmount = newLimit*(10**18);
}

function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {
    swapAndLiquifyEnabled = _enabled;
    emit SwapAndLiquifyEnabledUpdated(_enabled);
}
```

Contract owner can change max tx amount

```
function setMaxTxAmount(uint256 _maxBuyTxAmount, uint256 _maxSellTxAmount) public onlyOwner {
    maxBuyTransactionAmount = _maxBuyTxAmount * (10**18);
    maxSellTransactionAmount = _maxSellTxAmount * (10**18);
}
```

Contract owner can change the fees up to 100%

```
function setBuyFees(uint256 _buybackFee, uint256 _marketingFee, uint256 _charityFee, uint256 _devFee)
public onlyOwner {
    buyBackBuyFee = _buybackFee;
    marketingBuyFee = _marketingFee;
    charityBuyFee = _charityFee;
    devBuyFee = _devFee;
    totalBuyFees = buyBackBuyFee.add(marketingBuyFee).add(charityBuyFee).add(devBuyFee);
}

function setSellFees(uint256 _buybackFee, uint256 _marketingFee, uint256 _charityFee, uint256 _devFee)
public onlyOwner {
    buyBackSellFee = _buybackFee;
    marketingSellFee = _marketingFee;
    charitySellFee = _charityFee;
    devSellFee = _devFee;
    totalSellFees = buyBackSellFee.add(marketingSellFee).add(charitySellFee).add(devSellFee);
}
```

Contract owner can change buyback settings

```
function SetBuyBackUpperLimit(uint256 newLimit) external onlyOwner {
    buyBackUpperLimit = newLimit;
}

function setBuyBackEnabled(bool _enabled) public onlyOwner {
    buyBackEnabled = _enabled;
    emit BuyBackEnabledUpdated(_enabled);
}

function setBuyBackDivisor(uint256 _newValue) external onlyOwner() {
    require(_newValue > 0, "cannot be set as zero");
    _buyBackDivisor = _newValue;
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    _setOwner(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");
    _setOwner(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:	10%
Sell fees:	15%
Max TX:	1,000,000,000,000
Max Sell:	1,000,000,000,000

Honeypot Risk

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

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean



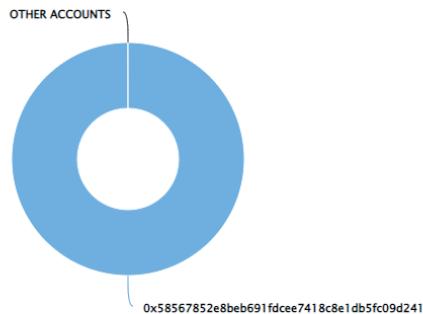
DRAGON INU TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (2,000,000,000,000.00 Tokens) of Dragon Inu

Token Total Supply: 2,000,000,000,000.00 Token | Total Token Holders: 1

Dragon Inu Top 10 Token Holders

Source: BscScan.com



(A total of 2,000,000,000,000.00 tokens held by the top 10 accounts from the total supply of 2,000,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0x58567852e8beb691fdcee7418c8e1db5fc09d241	2,000,000,000,000	100.0000%

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.

