



## SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Def.Cafe  
\$CAFE



02/02/2022



# TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-6 OWNER PRIVILEGES
- 7 CONCLUSION AND ANALYSIS
- 8 TOKEN DETAILS
- 9 DEF.CAFE TOKEN DISTRIBUTION & 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 Def.Cafe (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xdefCafE7eAC90d31BbBA841038DF365DE3c4e207

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 02/02/2022



# 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 exclude/include wallet from fees

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

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

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;
}
```

Contract owner can change swap settings

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

Contract owner can change fees up to 20%

```
function setFee(uint256 fee) public onlyOwner {
    require(fee >= 5 && fee <= 20, "Fee cannot exceed 20 percent.");
    require(threeDaysFeePeriodStarted && block.timestamp >= feePeriodEndTime, "Fee cannot be changed during 3 days after launch.");
    _Fee = fee;
}
```

## Contract owner can change `_developerWallet` and `_rewardsWallet` address

Current addresses:

`_developerWallet`: 0xcaFe1111BeE7e3e14bbF6B399aF6a85971F4Ecc

`_rewardsWallet`: 0xcafe0100f80138f2f3c541a3d882e39e2e366650

```
function setDeveloperAddress(address adr) public onlyOwner {
    _developerWallet = adr;
}

function setRewardsWallet(address adr) public onlyOwner {
    _rewardsWallet = adr;
}
```

Once contract owner call `startThreeDaysPeriod` function, total fees would be:

\*First 10 minutes: 15% buy fee & 30% sell fee

After 10 minutes: 0% buy fee & 30% sell fee

After 5 days: 0% buy fee & 15% sell fee

```
function startThreeDaysPeriod() public onlyOwner {
    require(!threeDaysFeePeriodStarted, "Already started.");
    feePeriodStartTime = block.timestamp;
    feePeriodEndTime = feePeriodStartTime.add(4 * 1 days);
    buyingFeePeriodEndTime = block.timestamp.add(10*1 minutes);
    threeDaysFeePeriodStarted = true;
}
```

```
function getCurrentFee() public view returns(uint256) {
    uint256 resultFee = _Fee;
    if(threeDaysFeePeriodStarted && block.timestamp <= feePeriodEndTime) {
        uint256 daysLeft = (feePeriodEndTime.sub(block.timestamp)).div(1 days);
        uint256 addedFee = daysLeft.mul(5);
        if(addedFee>15) addedFee = 15;
        resultFee = _Fee.add(addedFee);
    }
    return(resultFee);
}
```

\*Once contract owner call `startThreeDaysPeriod` function, buy fee would be 15% for 10 minutes

```
...
bool isBuyingFee = false;
bool isEnableFee = false;
if (_automatedMarketMakerPairs[to] && !_excluded[to] && !_excluded[from]) {
    isEnableFee = true;
} else if(block.timestamp <= buyingFeePeriodEndTime && _automatedMarketMakerPairs[from] && !_excluded[from] && !_excluded[to]) {
    isEnableFee = true;
    isBuyingFee = true;
}
...
```

# 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: 0%

Sell fees: 15%

Max TX: N/A

Max Sell: N/A

## Honeypot Risk

Ownership: Owned

Blacklist: Not detected

Modify Max TX: Not detected

Modify Max Sell: Not detected

Disable Trading: Not detected

## Rug Pull Risk

Liquidity: N/A

Holders: Clean



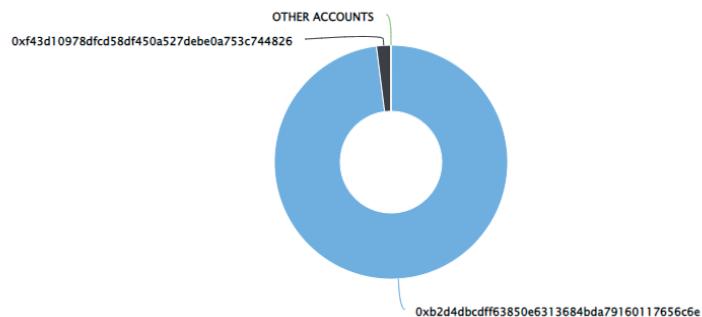
# DEF.CAFE TOKEN DISTRIBUTION & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (10,000,000.00 Tokens) of Def.Cafe

Token Total Supply: 10,000,000.00 Token | Total Token Holders: 2

Def.Cafe Top 10 Token Holders

Source: BscScan.com



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

Rank	Address	Quantity (Token)	Percentage
1	0xb2d4dbcdff63850e6313684bda79160117656c6e	9,799,998.72	98.0000%
2	0xf43d10978dfcd58df450a527debe0a753c744826	200,001.28	2.0000%

1. 0xb2d4dbcdff63850e6313684bda79160117656c6e - PinkSale presale address
2. 0xf43d10978dfcd58df450a527debe0a753c744826 - Team Tokens

# 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.

