



# SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



SOULCIETY  
CLUB

**Soulciety Club**  
\$SOUL

**25/05/2022**

# TABLE OF CONTENTS

- 1 **DISCLAIMER**
- 2 **INTRODUCTION**
- 3-4 **AUDIT OVERVIEW**
- 5-7 **OWNER PRIVILEGES**
- 8 **CONCLUSION AND ANALYSIS**
- 9 **TOKEN DETAILS**
- 10 **SOULCIETY CLUB TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS**
- 11 **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 Honeygot etc )



# INTRODUCTION

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

**0xd4466532583728a8516578AC5c41EC70b7af29aF**

**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 **25/05/2022**



# 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	Low
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 tax

```
function changelsFeeExempt(address holder, bool exempt) external authorized {
    isFeeExempt[holder] = exempt;
}
```

## Contract owner can exclude/include wallet from dividends

```
function changelsDividendExempt(address holder, bool exempt) external authorized {
    require(holder != address(this) && holder != pair);
    isDividendExempt[holder] = exempt;

    if(exempt){
        dividendDistributor.setShare(holder, 0);
    }else{
        dividendDistributor.setShare(holder, _balances[holder]);
    }
}
```

## Contract owner can exclude/include wallet from tx limitations

```
function changelsTxLimitExempt(address holder, bool exempt) external authorized {
    isTxLimitExempt[holder] = exempt;
}
```

## Contract owner can exclude/include address from wallet limitations

```
function changeRestrictWhales(bool newValue) external authorized {
    restrictWhales = newValue;
}
```

## Contract owner can change swap settings

```
function changeSwapBackSettings(bool enableSwapBack, uint256 newSwapBackLimit, bool swapByLimitOnly) external authorized {
    swapAndLiquifyEnabled = enableSwapBack;
    swapThreshold = newSwapBackLimit;
    swapAndLiquifyByLimitOnly = swapByLimitOnly;
}
```

## Contract owner can change tx limitations

```
function changeTxLimit(uint256 newLimit) external authorized {
    _maxTxAmount = newLimit;
    require(_maxTxAmount >= _totalSupply/100);
}
```

## Contract owner can change wallet limitations

```
function changeWalletLimit(uint256 newLimit) external authorized {
    _walletMax = newLimit;
    require(_walletMax >= _totalSupply/100);
}
```

## Contract owner can exclude/include wallet(s) from transactions

```
function enable_blacklist(bool _status) public onlyOwner {
    blacklistMode = _status;
}

function manage_blacklist(address[] calldata addresses, bool status)
    public
    onlyOwner
{
    for (uint256 i; i < addresses.length; ++i) {
        isBlacklisted[addresses[i]] = status;
    }
}
```

## Contract owner can change reward token

```
function setRewardToken(address _rewardToken) external authorized {
    dividendDistributor.setRewardToken(_rewardToken);
}
```

## Contract owner can change **autoLiquidityReceiver** and **developmentWallet** addresses

Current values:

**autoLiquidityReceiver** : 0x4f31428f30ef91bb73a16aa232e7c8ab013495d3

**developmentWallet** : 0x4f31428f30ef91bb73a16aa232e7c8ab013495d3

```
function changeFeeReceivers(address newLiquidityReceiver, address newMarketingWallet) external authorized {
    autoLiquidityReceiver = newLiquidityReceiver;
    developmentWallet = newMarketingWallet;
}
```



## Contract owner can change fees up to 25%

```
function changeFees(uint256 newLiqFee, uint256 newRewardFee, uint256 newMarketingFee) external
authorized {
    liquidityFee = newLiqFee;
    rewardsFee = newRewardFee;
    developmentFee = newMarketingFee;

    totalFee = liquidityFee.add(developmentFee).add(rewardsFee);
    totalFeelfSelling = totalFee;

    require(totalFee <=25,"Fees too high");
}
```

## Contract owner can burn tokens from specific wallet

```
function _burn(address account, uint256 amount) internal virtual {
    _balances[account] = _balances[account].sub(amount);
    _totalSupply -= amount;
    emit Transfer(account, address(0), amount);
}

function burn(uint256 amount) external {
    _burn(msg.sender, amount);
}
```

## Contract owner can transfer ownership

```
function transferOwnership(address payable adr) public onlyOwner {
    owner = adr;
    authorizations[adr] = true;
    emit OwnershipTransferred(adr);
}

event OwnershipTransferred(address owner);
```



# 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 1 LOW issue during the first review.

# TOKEN DETAILS

## Details

Buy fees:	10%
Sell fees:	10%
Max TX:	1,000,000,000
Max Sell:	N/A

## Honeypot Risk

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

## Rug Pull Risk

Liquidity:	N/A
Holders:	Clean



# Soulciety Club Token Token Analytics & Top 10 Token Holders



Rank	Address	Quantity (Token)	Percentage
1	0x4f31428f30ef91bb73a16aa232e7c8ab013495d3	1,000,000,000	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.

