freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Christmas Floki

$FLOC

14/11/2022

TOKEN OVERVIEW

Fees

• Buy fees:	9%
• Sell fees:	9%

Fees privileges

- Can change buy fees up to 20% and sell fees up to 20%

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change max tx amount and max wallet amount with threshold

Blacklist

- No blacklist function

Other privileges

- Can exclude / include from fees
- Can exclude / include from rewards

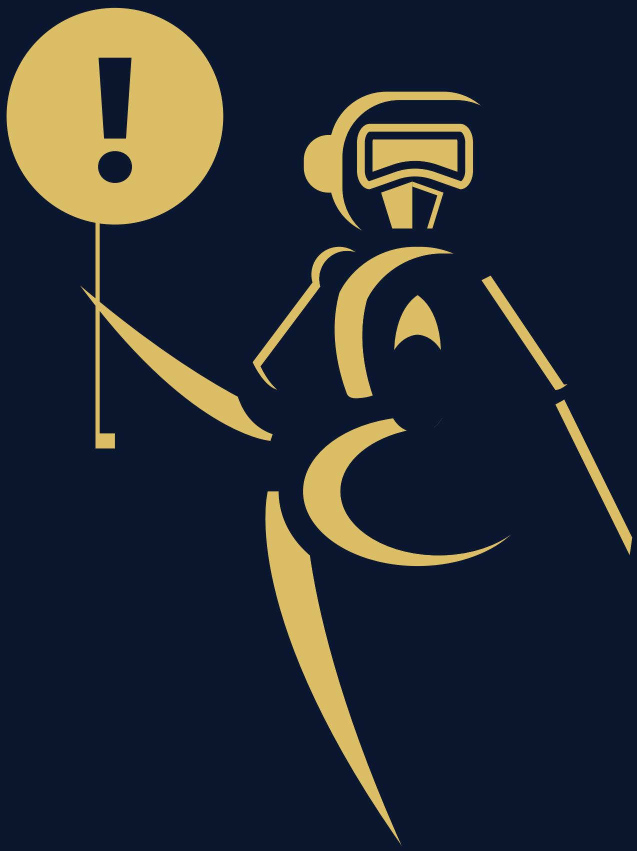
TABLE OF CONTENTS

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 Christmas Floki (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xe5765E33E349B2Dcf22A37b2b4E87c10ad43F165

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 14/11/2022

AUDIT OVERVIEW

91

Security Score

93

Static Scan

Automatic scanning for common vulnerabilities

90

ERC Scan

Automatic checks for ERC's conformance

0	High
2	Medium
1	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't mint tokens after initial contract deploy

● Contract owner can't exclude an address from transactions

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

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
      require(_isExcludedFromFees[account] != excluded, "FLOC: 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 dividends

```
function excludeFromDividends(address account) external onlyOwner {
            require(!excludedFromDividends[account]);
            excludedFromDividends[account] = true;
            _setBalance(account, 0);
            tokenHoldersMap.remove(account);
            emit ExcludeFromDividends(account);
}
```

● Contract owner can exclude/include wallet from tax and dividends

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

● Contract owner can enable/disable wallet limitations

```
function enableMaxWalletLimit() external onlyOwner {
      require(maxWalletLimit == false, "Already max wallet limit is enabled");
      maxWalletLimit = true;
}

function disableMaxWalletLimit() external onlyOwner {
      require(maxWalletLimit == true, "Already max wallet limit is disabled");
      maxWalletLimit = false;
}
```

Contract owner can change buy fees up to 20% and sell fees up to 20%

```
function setBuyFee(uint256 rewardFee, uint256 liquidityFee, uint256 marketingFee) external onlyOwner{
    buyBUSDRewardsFee = rewardFee;
    buyLiquidityFee = liquidityFee;
    buyMarketingFee = marketingFee;
    buyTotalFees = buyBUSDRewardsFee.add(buyLiquidityFee).add(buyMarketingFee);
    require(buyTotalFees <= 20, "Fees must be at 20% or less");
}

function setSellFee(uint256 rewardFee, uint256 liquidityFee, uint256 marketingFee) external onlyOwner{
    sellBUSDRewardsFee = rewardFee;
    sellLiquidityFee = liquidityFee;
    sellMarketingFee = marketingFee;
    sellTotalFees = sellBUSDRewardsFee.add(sellLiquidityFee).add(sellMarketingFee);
    require(sellTotalFees <= 20, "Fees must be at 20% or less");
}
```

Contract owner can change _marketingWalletAddress address

Current value:

_marketingWalletAddress : 0xf5d639c31dad288921fb7cc7506ab139fd5d38f2

```
function setMarketingWallet(address payable wallet) external onlyOwner{
    _marketingWalletAddress = wallet;
}
```

Contract owner can change max tx amount and max wallet limitations

```
function setMaxTxAmount(uint256 newAmount) external onlyOwner() {
    _maxTxAmount = newAmount * (10**18);
    require(_maxTxAmount >= totalSupply() * 5 / 1000, "cant set max trx less then 0.5%");

}

function setMaxWalletToken(uint256 _maxWallet) external onlyOwner {
    maxWallet = _maxWallet * (10**18);
    require(maxWallet >= totalSupply() * 5 / 1000, "cant set max wallet less then 0.5%");
 }
```

Contract owner can withdraw stuck BNB or tokens ($FLOC excluded)

```
function withdrawStuckBNB() external onlyOwner{
    require (address(this).balance > 0, "Can't withdraw negative or zero");
    payable(owner()).transfer(address(this).balance);
}

function removeStuckToken(address _address) external onlyOwner {
    require(_address != address(this), "Can't withdraw tokens destined for liquidity");
    require(IERC20(_address).balanceOf(address(this)) > 0, "Can't withdraw 0");

    IERC20(_address).transfer(owner(), IERC20(_address).balanceOf(address(this)));
}
```

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;
}
```

Recommendation:

The team should carefully manage the private keys of the owner's account. We strongly recommend a powerful security mechanism that will prevent a single user from accessing the contract admin functions. The risk can be prevented by temporarily locking the contract or renouncing ownership.

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 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	9%
Sell fees:	9%
Max TX:	1,000,000,000,000
Max Sell:	N/A

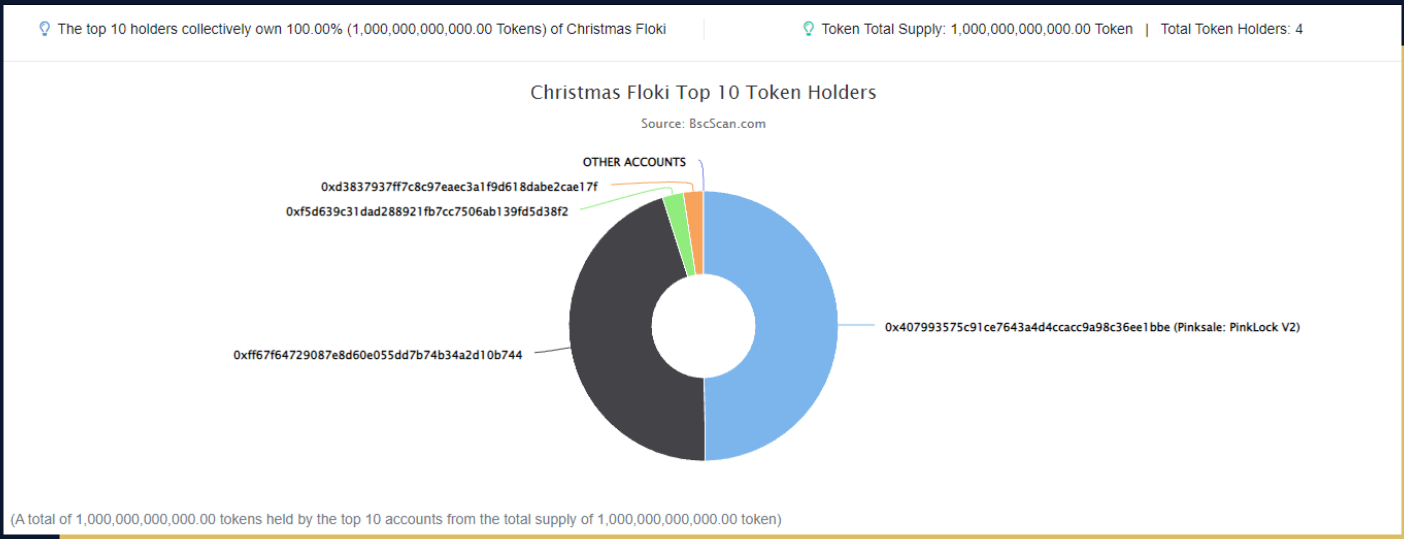
Honeypot Risk

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

Others

Liquidity:	N/A
Holders:	Clean

CHRISTMAS FLOKI TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



The top 10 holders collectively own 100.00% (1,000,000,000,000.00 Tokens) of Christmas Floki

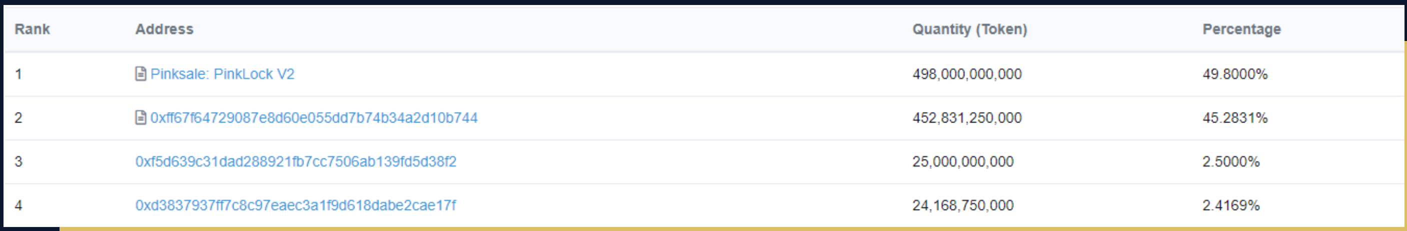
Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 4

Christmas Floki Top 10 Token Holders

Source: BscScan.com

OTHER ACCOUNTS
0xd3837937ff7c8c97eaec3a1f9d618dabe2cae17f
0xf5d639c31dad288921fb7cc7506ab139fd5d38f2
0x407993575c91ce7643a4d4ccacc9a98c36ee1bbe (Pinksale: PinkLock V2)
0xff67f64729087e8d60e055dd7b74b34a2d10b744

(A total of 1,000,000,000,000.00 tokens held by the top 10 accounts from the total supply of 1,000,000,000,000.00 token)



Rank	Address	Quantity (Token)	Percentage
1	Pinksale: PinkLock V2	498,000,000,000	49.8000%
2	0xff67f64729087e8d60e055dd7b74b34a2d10b744	452,831,250,000	45.2831%
3	0xf5d639c31dad288921fb7cc7506ab139fd5d38f2	25,000,000,000	2.5000%
4	0xd3837937ff7c8c97eaec3a1f9d618dabe2cae17f	24,168,750,000	2.4169%

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.