freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

PinkCat Token

$PCT

14/03/2023

TOKEN OVERVIEW

Fees

- Buy fees: 3%
- Sell fees: 3%

Fees privileges

- Can change fees up to 10%

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change / set max tx amount or max wallet amount (with threshold)

Blacklist

- Blacklist function not detected

Other privileges

- Can enable / disable fees

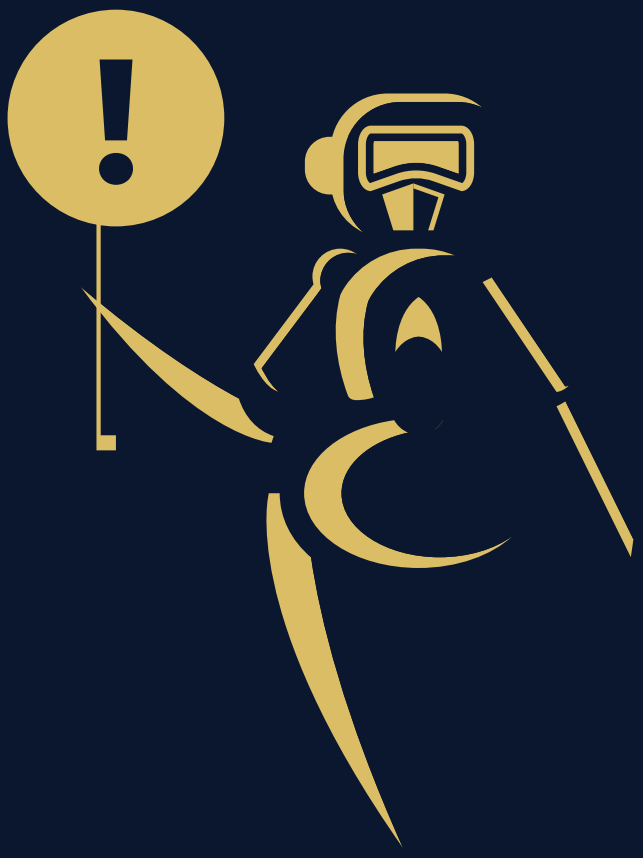
TABLE OF CONTENTS

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 PinkCat Token (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x8463453E428DF29048Db266933456049Ca1A1cE3

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 14/03/2023

WEBSITE DIAGNOSTIC

https://www.pinkcat.io/

0-49 50-89 90-100

Performance Accessibility Best Practices SEO Progressive Web App

Socials

Twitter

https://twitter.com/playthepink

Telegram

https://t.me/playthepink

AUDIT OVERVIEW

88

Security Score

94

Static Scan
Automatic scanning for common vulnerabilities

84

ERC Scan
Automatic checks for ERC's conformance

0 High

2 Medium

0 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 include / exclude wallet from tax

```
function setExcludedFromFee(address account, bool e) external onlyOwner {
    _isExcludedFromFee[account] = e;
}
```

- Contract owner can include / exclude wallet from tax

```
function excludeFromReward(address account) external 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 excluded");
    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 _marketingWallet address

Current value:

_marketingWallet: 0xbb05e2788ce696c399e627f1a953c80d119196c5

```
function setMarketingWallet(address marketingWallet) external onlyOwner {
    _marketingWallet = marketingWallet;
}
```

Contract owner has to call enableTrading function to enable trade

once enabled, can't be disabled

```
function enableTrading() external onlyOwner {
    require(!tradingActive, "Trading is already enabled");
    tradingActive = true;
    swapEnabled = true;
}
```

Contract owner can withdraw tokens from smart contract

except own tokens

```
function rescueBNB(uint256 weiAmount) external onlyOwner {
    payable(owner()).transfer(weiAmount);
}

function rescueBSC20(address tokenAdd, uint256 amount) external onlyOwner {
    require(tokenAdd != address(this), "Owner can't claim contract's balance of its own tokens");
    IBEP20(tokenAdd).transfer(owner(), amount);
}
```

Contract owner can remove all limits (tradingActive, maxTxAmount,maxWalletToken)

```
function removeLimits() external onlyOwner {
    limitsInEffect = false;
}
```

Contract owner can change tax up to 10%

```
function setTaxFeePercent(uint256 amount) external onlyOwner {
    require(amount <= 4, "Holder Reflection cannot exceed 4%");
    taxFee = amount;
}

function setLiquidityFeePercent(uint256 amount) external onlyOwner {
    require(amount <= 6, "Liquidity Fee cannot exceed 6%");
    liquidityFee = amount;
}
```

Contract owner can change max tx amount and max wallet (with threshold)

```
function setMaxWalletTokens(uint256 amount) external onlyOwner {
    require(amount >= _tTotal / 100, "Cannot set maxTransactionAmount lower than 1%");
        maxWalletToken = amount ;
}

function setMaxTxAmount(uint256 amount) external onlyOwner {
    require(amount >= _tTotal * 5 / 1000, "Cannot set maxTransactionAmount lower than 0.5%");
    maxTxAmount = amount;
}
```

- Contract owner can change swap settings

```
function setSwapAndLiquifyEnabled(bool enabled) external onlyOwner {
        swapEnabled = enabled;
        emit SwapAndLiquifyEnabledUpdated(enabled);
    }
```

- Contract owner can transfer ownership

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

    function _transferOwnership(address newOwner) internal virtual {
        address oldOwner = _owner;
        _owner = newOwner;
        emit OwnershipTransferred(oldOwner, newOwner);
    }
```

- Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
        _transferOwnership(address(0));
    }
```

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:	3%
Sell fees:	3%
Max TX:	100,000
Max Sell:	N/A

Honeypot Risk

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

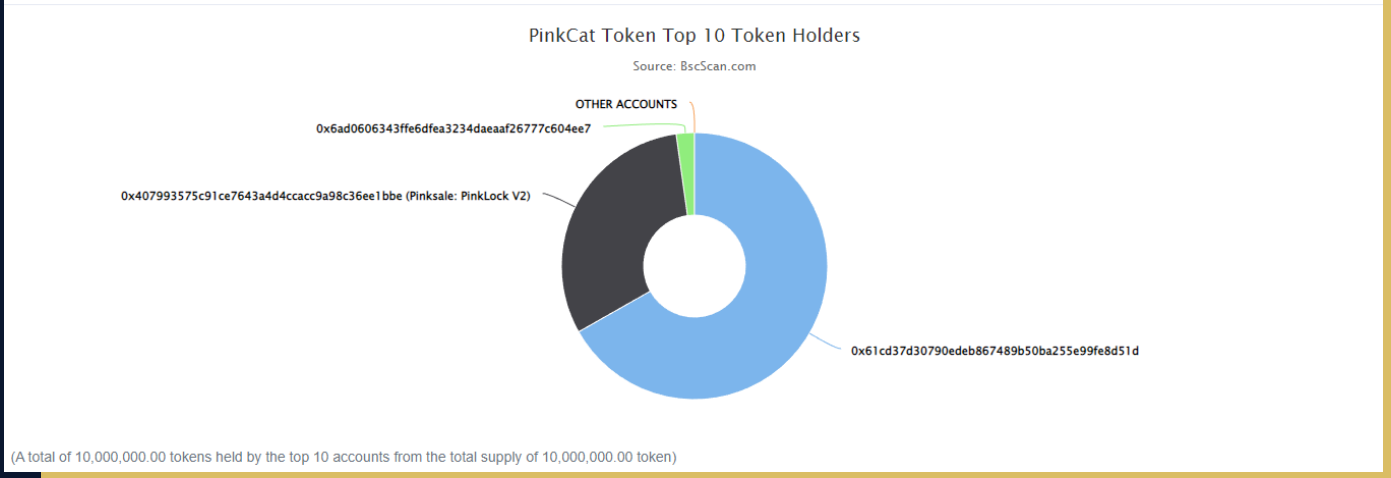
Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

PCT TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (10,000,000.00 Tokens) of PinkCat Token

Token Total Supply: 10,000,000.00 Token | Total Token Holders: 3



(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	0x61cd37d30790edeb867489b50ba255e99fe8d51d	6,680,250	66.8025%
2	Pinksale: PinkLock V2	3,100,000	31.0000%
3	0x6ad0606343ffe6dfea3234daeaaf26777c604ee7	219,750	2.1975%

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.