freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

CatJump
$CATJUMP

23/03/2022

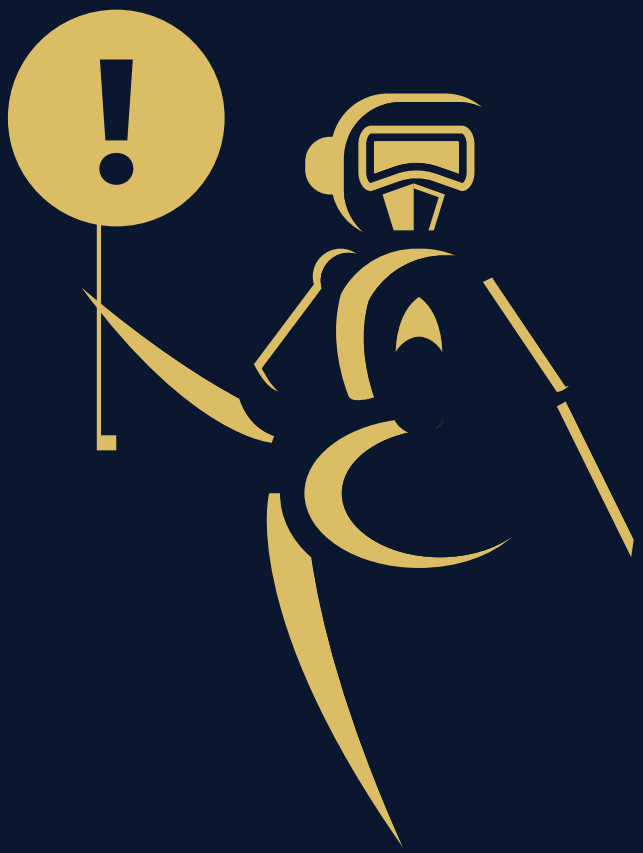
TABLE OF CONTENTS

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

0xcc0ce161a6505311734b078Cf44D16d204E9416A

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 23/03/2022

AUDIT OVERVIEW

96

Security Score

94

Static Scan
Automatic scanning for common vulnerabilities

98

ERC Scan
Automatic checks for ERC's conformance

0 High

0 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 exclude/include wallet from tax

```
function excludeFromFee(address account) external onlyOwner {
        _isExcludedFromFee[account] = true;
}

function includeInFee(address account) external onlyOwner {
        _isExcludedFromFee[account] = false;
}
```

Contract owner can exclude/include wallet from reward

```
function excludeFromReward(address account) public 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 exclude/include wallet from tx limitations

```
function excludeFromTxLimit(address account, bool _value) external onlyOwner {
        _isExemptFromTxLimit[account] = _value;
}

function setExcludedFromWhale(address account, bool _enabled) public onlyOwner  {
        _isExcludedFromWhale[account] = _enabled;
}
```

Contract owner can change fees (with threshold)

```
function setAllBuyFeePercent(uint256 liquidityFee, uint256 marketingFee, uint256 ecosystemFee) external
onlyOwner()
  {
      _liquidityFee = liquidityFee;
      _previousLiquidityFee = liquidityFee;

      _marketingFee = marketingFee;
      _previousMarketingFee = marketingFee;

      _ecosystemFee = ecosystemFee;
      _previousEcosystemFee = ecosystemFee;

      require(_liquidityFee.add(_marketingFee).add(_ecosystemFee)<=30, "Too High Fee");

}


function setAllSaleFeePercent(uint256 liquidityFee, uint256 marketingFee, uint256 ecosystemFee) external
onlyOwner()
  {
      _saleLiquidityFee = liquidityFee;
      _saleMarketingFee = marketingFee;
      _saleEcosystemFee = ecosystemFee;
      require(_saleLiquidityFee.add(_saleMarketingFee).add(_saleEcosystemFee)<=40, "Too High Fee");
}
```

Contract owner can change swap settings

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

Contract owner can change marketingAddress and ecosystemAddress addresses

Current values:

marketingAddress : 0x5fe57ce813a8e7d7e2597c1f4af24468f0faa9ae

ecosystemAddress : 0xa45e72d049bcaa046a76439547a820521a4c663d

```
function setMarketingAddress(address _marketingAddress) external onlyOwner() {
      marketingAddress = _marketingAddress;
}

function setEcosystemAddress(address _ecosystemAddress) external onlyOwner() {
      ecosystemAddress = _ecosystemAddress;
}
```

Contract owner can set max buy and sell limit (with threshold)

```
function setMaxBuyTxAmount(uint256 _mount) external onlyOwner() {
        require(_mount>_tTotal.div(10000), "Too low Txn limit"); // Min 0.1%
        _maxBuyTxAmount = _mount;
}

function setMaxSaleTxAmount(uint256 _mount) external onlyOwner() {
        require(_mount>_tTotal.div(10000), "Too low Txn limit"); // Min 0.1%
        _maxSaleTxAmount = _mount;
}
```

Contract owner can set wallet limitation (with threshold)

```
function  setWalletMaxHoldingLimit(uint256 _amount) public onlyOwner  {
          _walletHoldingMaxLimit = _amount;
          require(_amount>_tTotal.div(10000), "Too less limit");
}
```

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;
}
```

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:	7%
Sell fees:	12%
Max TX:	5,000,000,000
Max Sell:	5,000,000,000

Honeypot Risk

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

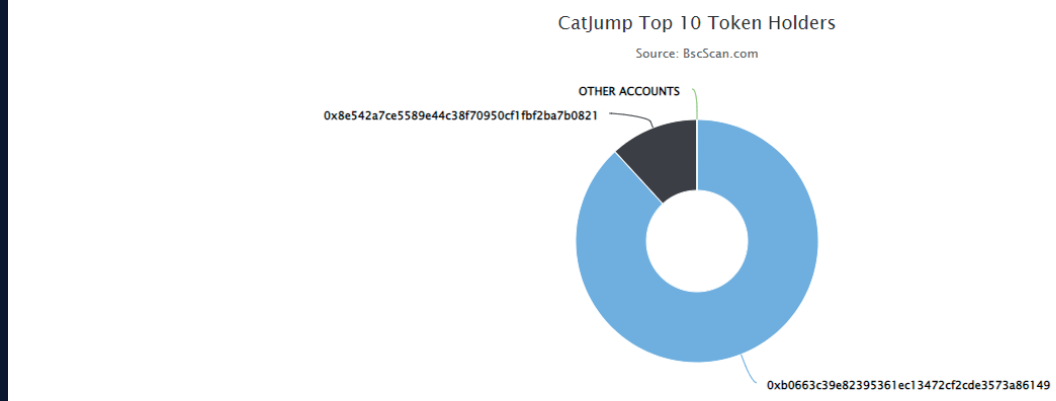
Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

CATJUMP TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (100,000,000,000.00 Tokens) of CatJump

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 2



(A total of 100,000,000,000.00 tokens held by the top 10 accounts from the total supply of 100,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0xb0663c39e82395361ec13472cf2cde3573a86149	88,189,200,000	88.1892%
2	0x8e542a7ce5589e44c38f70950cf1fbf2ba7b0821	11,810,800,000	11.8108%

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.