freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

CATZ INU
$CATZU

21/04/2023

TOKEN OVERVIEW

Fees

• **Buy fees:**   10%

• **Sell fees:**   10%

• **Transfer fees:**   0%

Fees privileges

• **Can change fees up to 25%**

Ownership

• **Owned**

Minting

• **No mint function**

Max Tx Amount / Max Wallet Amount

• **Can't change max tx amount and/or max wallet amount**

Blacklist

• **Blacklist function not detected**

Other privileges

• **Can exclude / include from 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
CATZ INU (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x67FfF590acA5141f0C790b3284334CAe15b73770

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 **21/04/2023**

WEBSITE DIAGNOSTIC

https://www.catzinu.io/

0-49   **50-89**   **90-100**

91
Performance

90
Accessibility

91
Best
Practices

92
SEO

NA
Progressive
Web App

Socials

Twitter

https://twitter.com/CatZCoin

Telegram

https://t.me/CatZinuofficial

AUDIT OVERVIEW

92

Security Score

98

Static Scan
Automatic scanning for common vulnerabilities

90

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 exclude/include wallet from tax**

```
function excludeFromFees(address account, bool excluded) external onlyOwner {
    require(_isExcludedFromFees[account] != excluded, "Account is already set to that state");
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}
```

● **Contract owner can exclude 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 change swap settings**

```
function setSwapTokensAtAmount(uint256 newAmount) external onlyOwner{
    require(newAmount > totalSupply() / 100_000, "SwapTokensAtAmount must be greater than 0.001% of
total supply");
    swapTokensAtAmount = newAmount;
}
```

● **Contract owner can withdraw tokens from smart contract**
(native tokens excluded)

```
function claimStuckTokens(address token) external onlyOwner {
    require(token != address(this), "Owner cannot claim native tokens");
    if (token == address(0x0)) {
        payable(msg.sender).transfer(address(this).balance);
        return;
    }
    IERC20CATZ ERC20token = IERC20CATZ(token);
    uint256 balance = ERC20token.balanceOf(address(this));
    ERC20token.transfer(msg.sender, balance);
}
```

● Contract owner can change fees up to 25%

```solidity
function updateFees(
    uint256 _liquidityFeeOnBuy,
    uint256 _devFeeOnBuy,
    uint256 _marketingFeeOnBuy,
    uint256 _rewardsFeeOnBuy,
    uint256 _liquidityFeeOnSell,
    uint256 _devFeeOnSell,
    uint256 _marketingFeeOnSell,
    uint256 _rewardsFeeOnSell,
    uint256 _liquidityFeeOnTransfer,
    uint256 _devFeeOnTransfer,
    uint256 _marketingFeeOnTransfer,
    uint256 _rewardsFeeOnTransfer
) external onlyOwner {
    liquidityFeeOnBuy  = _liquidityFeeOnBuy;
    devFeeOnBuy        = _devFeeOnBuy;
    marketingFeeOnBuy  = _marketingFeeOnBuy;
    rewardsFeeOnBuy    = _rewardsFeeOnBuy;

    totalBuyFee = _liquidityFeeOnBuy + _devFeeOnBuy + _marketingFeeOnBuy + _rewardsFeeOnBuy;

    liquidityFeeOnSell  = _liquidityFeeOnSell;
    devFeeOnSell        = _devFeeOnSell;
    marketingFeeOnSell  = _marketingFeeOnSell;
    rewardsFeeOnSell    = _rewardsFeeOnSell;

    totalSellFee = _liquidityFeeOnSell + _devFeeOnSell + _marketingFeeOnSell + _rewardsFeeOnSell;

    liquidityFeeOnTransfer  = _liquidityFeeOnTransfer;
    devFeeOnTransfer        = _devFeeOnTransfer;
    marketingFeeOnTransfer  = _marketingFeeOnTransfer;
    rewardsFeeOnTransfer    = _rewardsFeeOnTransfer;

    totalTransferFee = _liquidityFeeOnTransfer + _devFeeOnTransfer + _marketingFeeOnTransfer + _rewardsFeeOnTransfer;

    require((totalBuyFee + totalSellFee + totalTransferFee) <= 25, "Buy fee cannot be more than 25%");

    emit FeesUpdated(totalBuyFee, totalSellFee, totalTransferFee);
}
```

● Contract owner can change marketingAddress address

Current value:

marketingAddress : 0xb1c994197e0f4678c7e54a58ca6a3c37f7da1381

```solidity
function changeMarketingWallet(address _marketingAddress) external onlyOwner {
    require(_marketingAddress != marketingAddress, "Marketing wallet is already that address");
    require(!isContract(_marketingAddress), "Marketing wallet cannot be a contract");
    require(_marketingAddress != DEAD, "Marketing wallet cannot be the zero address");
    marketingAddress = _marketingAddress;
    emit MarketingWalletChanged(marketingAddress);
}
```

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:	10%
Sell fees:	10%
Transfer fees:	0%
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

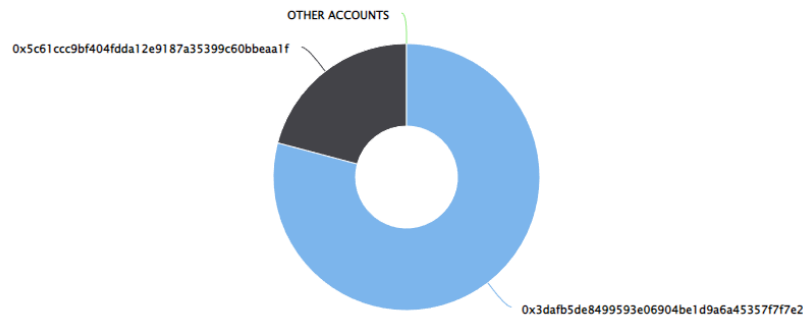|	
Liquidity:	N/A
Holders:	20,87% tokens unlocked

CATZU TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (1,000,000,000.00 Tokens) of CATZ INU | Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 2

CATZ INU Top 10 Token Holders

Source: BscScan.com

OTHER ACCOUNTS

0x5c61ccc9bf404fdda12e9187a35399c60bbeaa1f

0x3dafb5de8499593e06904be1d9a6a45357f7f7e2

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

Rank	Address	Quantity (Token)	Percentage
1	0x3dafb5de8499593e06904be1d9a6a45357f7f7e2	791,280,000	79.1280%
2	0x5c61ccc9bf404fdda12e9187a35399c60bbeaa1f	208,720,000	20.8720%

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.