



# SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



**CATBOME**

\$CATBOME

**05/04/2024**



# TOKEN OVERVIEW

---

## Fees

- Buy fees: 10%
- Sell fees: 10%

## Fees privileges

- Can't change fees

## 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
  - Contract owner has the ability to update the dev wallet without requiring any additional checks
-

# TABLE OF CONTENTS

1

DISCLAIMER

2

INTRODUCTION

3

WEBSITE + SOCIALS

4-5

AUDIT OVERVIEW

6-7

OWNER PRIVILEGES

8

CONCLUSION AND ANALYSIS

9

TOKEN DETAILS

10

CATBOME 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 Honeypot etc )



# INTRODUCTION

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

**0x8bCa9040A7d0221beB2D97cFF37E3ecE3757a6f2**

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 **05/04/2024**



# WEBSITE DIAGNOSTIC

<https://catbome.io/>



0-49



50-89



90-100



Performance



Accessibility



Best  
Practices



SEO



Progressive  
Web App

## Socials



Twitter

[https://x.com/cat\\_bome](https://x.com/cat_bome)



Telegram

N/A

# AUDIT OVERVIEW



Security Score  
**HIGH RISK**  
Audit FAIL



**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	Low
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't exclude addresses from transactions

## ● Contract owner can change dev wallet address

Since the function lacks additional checks, the contract owner could potentially set the dev wallet to a contract address that is unable to receive ETH/BNB, leading to the transformation of the token contract into a honeypot. This scenario would render selling impossible for users

Default value: `0xF694cC0702277E3354670063A64b3041ba2B3003`

```
function setUpdateDevAddress(address payable devAddress) public onlyOwner {
    emit setDevAddress(_devAddress, devAddress);
    _devAddress = devAddress;
    _isExcludedFromFee[_devAddress] = true;
}
```

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

```
function excludeFromFee(address add, bool excluded) public onlyOwner {
    _isExcludedFromFee[add] = excluded;
}

function excludeListFromFee(address[] calldata list, bool excluded) public onlyOwner {
    for(uint256 i = 0; i < list.length; i++) {
        _isExcludedFromFee[list[i]] = excluded;
    }
}
```

## ● Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    _transferOwnership(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");
    _transferOwnership(newOwner);
}

function _transferOwnership(address newOwner) internal virtual {
    address oldOwner = _owner;
    _owner = newOwner;
    emit OwnershipTransferred(oldOwner, 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 1 HIGH issues during the first review.

# TOKEN DETAILS

## Details

Buy fees:	10%
Sell fees:	10%
Max TX:	N/A
Max Wallet:	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:	100% unlocked tokens



# CATBOME TOKEN ANALYTICS

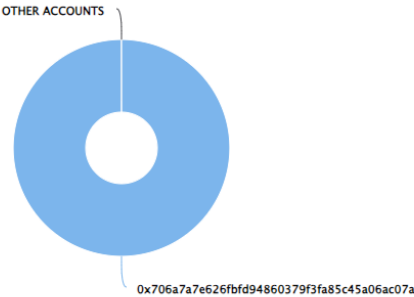
## & TOP 10 TOKEN HOLDERS

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

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

### CATBOME Top 10 Token Holders

Source: BscScan.com



(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	0x706A7A7E...5a06Ac07A	1,000,000,000,000	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.

