freshcoins

# SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

## OneBot

$ONE

13/09/2023

# TOKEN OVERVIEW

## Fees

- Buy fees: 2%
- Sell fees: 2%

## Fees privileges

- Can change buy fees up to 25% and sell fees up to 25%

## Ownership

- Owned

## Minting

- No mint function

## Max Tx Amount / Max Wallet Amount

- Can’t change max tx amount and max wallet amount

## Blacklist

- Blacklist function not detected

## Other privileges

- Can exclude / include from fees
- Contract owner has to call enableTrading function to enable trade

# TABLE OF CONTENTS

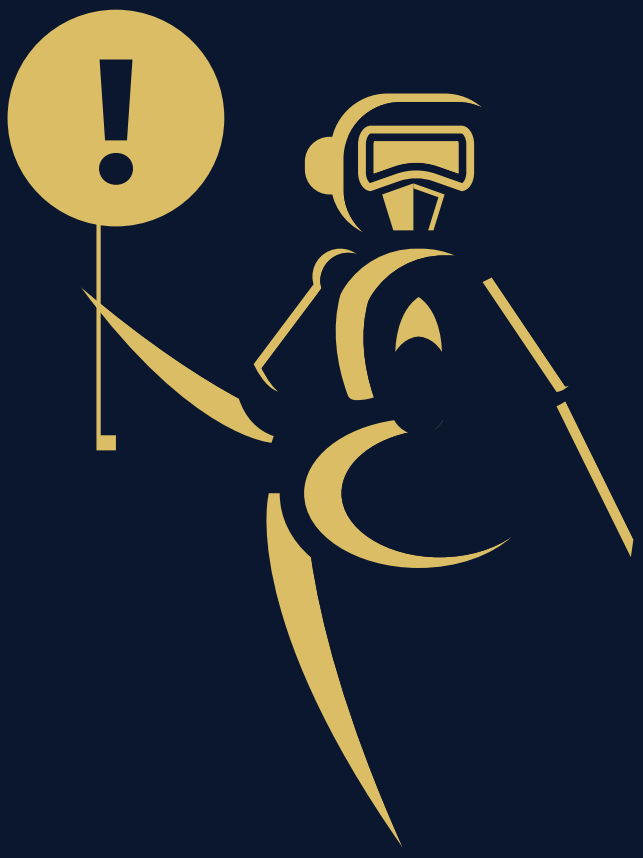


# DISCLAIMER

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

0x3E23187C077976D32D87D2ee998502b5b4836044

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 13/09/2023

# WEBSITE DIAGNOSTIC

https://one-bot.xyz/

0-49 50-89 90-100

| Performance | Accessibility | Best Practices | SEO | Progressive Web App |
|---|---|---|---|---|
| 92 | 94 | 91 | 90 | NA |

## Socials

Twitter

https://twitter.com/onebot_bsc

Telegram

https://t.me/onebot_bsc

# AUDIT OVERVIEW

82

Security Score

98

**Static Scan**
Automatic scanning for common vulnerabilities

80

**ERC Scan**
Automatic checks for ERC's conformance

1 High

0 Medium

0 Low

1 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 setIsFeeExempt(address holder, bool exempt) external onlyOwner {
        isFeeExempt[holder] = exempt;
        emit SetIsFeeExempt(holder, exempt);
}
```

- Contract owner has to call `enableTrading` function to enable trade

Please note that both the contract owner and any authorized wallet holders retain the ability to engage in trading, even in situations where trading has been disabled

```
function enableTrading() external onlyOwner {
        require(!isTradeEnabled, "Trading already enabled");
        isTradeEnabled = true;
        listingTime = block.number;
}

_transferFrom function line 533
.
.
.
if (!isTradeEnabled) require(isAuthorized[sender], "Trading disabled");
.
.
.
```

- Contract owner has the authority to grant permission for wallets to engage in trading even when trading is disabled

```
function setAuthorizedWallets(
        address _wallet,
        bool _status
    ) external onlyOwner {
        isAuthorized[_wallet] = _status;
}

_transferFrom function line 533
.
.
.
if (!isTradeEnabled) require(isAuthorized[sender], "Trading disabled");
.
.
.
```

## Contract owner has ability to retrieve BNB held by the contract

```
function rescueETH() external onlyOwner {
    uint256 balance = address(this).balance;
    require(balance > 0, "No enough ETH to transfer");

    payable(msg.sender).transfer(balance);
}
```

## Contract owner can change marketingWallet address

**Current value:**

**marketingWallet: 0xC6E5B70A4Da397951c6711496bb5eF22bfB0100B**

```
function changeMarketingWallet(address _wallet) external onlyOwner {
    marketingWallet = _wallet;
}
```

## Contract owner can change swap settings

```
function setDoContractSwap(bool _enabled) external onlyOwner {
    contractSwapEnabled = _enabled;

    emit SetDoContractSwap(_enabled);
}
```

## Contract owner can change buy fees up to 25% and sell fees up to 25%

```
function changeSellFees(uint256 _sellTotalFee) external onlyOwner {
    sellTotalFee = _sellTotalFee;

    require(sellTotalFee <= 25, "can not greater than 25%");
}

function changeBuyFees(uint256 _buyTotalFees) external onlyOwner {
    buyTotalFee = _buyTotalFees;

    require(buyTotalFee <= 25, "can not greater than 25%");
}
```

## 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));
}
```

## Missing Zero Address Check

- OneBot::changeMarketingWallet()

**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: | 2% |
| Sell fees: | 2% |
| 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 |

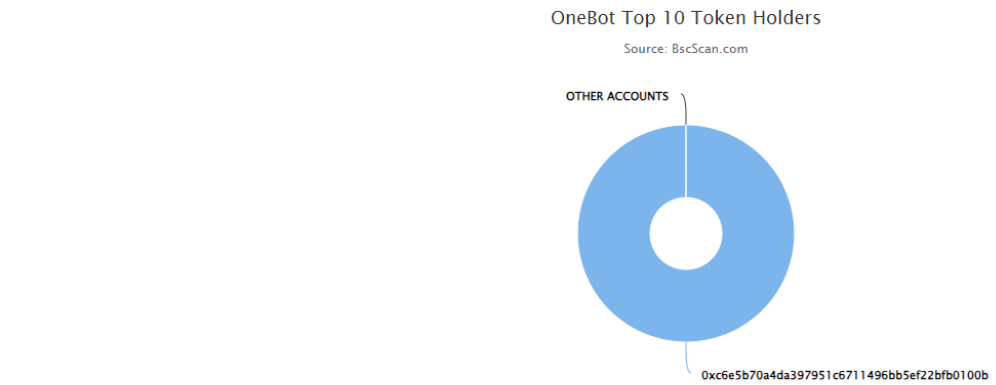
## Rug Pull Risk

| | |
|---|---|
| Liquidity: | N/A |
| Holders: | 100% unlocked tokens |

# ONE TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (4,000,000.00 Tokens) of OneBot

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




(A total of 4,000,000.00 tokens held by the top 10 accounts from the total supply of 4,000,000.00 token)

| Rank | Address | Quantity (Token) | Percentage |
|---|---|---|---|
| 1 | 0xC6E5B7...bfB0100B | 4,000,000 | 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.