freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

OlympicDogeBsc

$OlympicDoge

19/06/2023

TOKEN OVERVIEW

Fees

• Buy fees:	10%
• Sell fees:	10%

Fees privileges

• Can change buy fees up to 99% and sell fees up to 99%

Ownership

• Owned

Minting

• No mint function

Max Tx Amount / Max Wallet Amount

• Can change max tx amount and / or max wallet amount (with threshold)

Blacklist

• Blacklist function not detected

Other privileges

• Contract owner has to call swapTrading function to enable trade

• Can exclude / include from fees

TABLE OF CONTENTS

1 DISCLAIMER

2 INTRODUCTION

3 WEBSITE + SOCIALS

4-5 AUDIT OVERVIEW

6-9 OWNER PRIVILEGES

10 CONCLUSION AND ANALYSIS

11 TOKEN DETAILS

12 OLYMPICDOGE TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

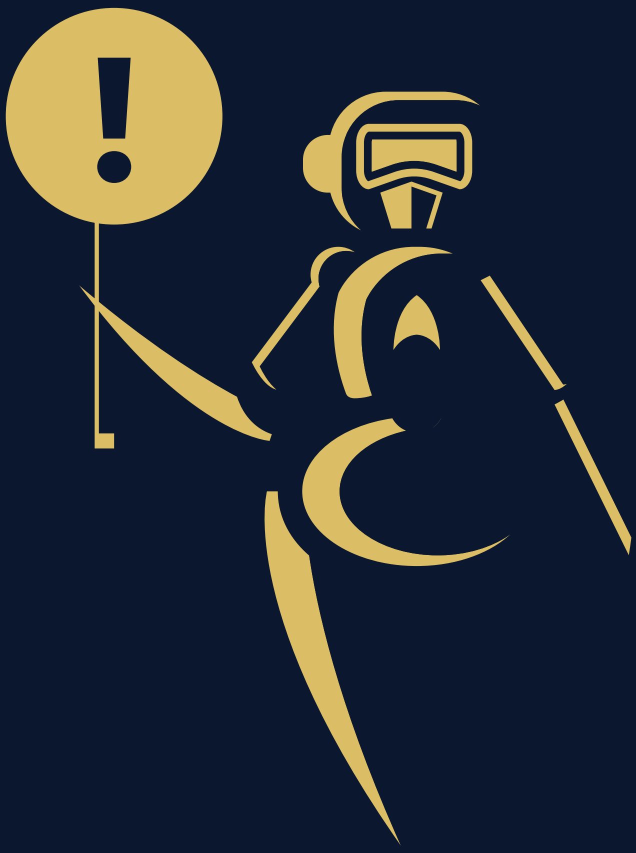
13 TECHNICAL DISCLAIMER

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

0xdF50C3d2623c9A6AD566Bfe22677Db7633888693

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 19/06/2023

WEBSITE DIAGNOSTIC

https://olympicdoge.finance/

🟥 0-49	🟨 50-89	🟩 90-100

Performance	Accessibility	Best Practices	SEO	Progressive Web App
91	93	91	90	NA

Socials

Twitter

https://twitter.com/OlympicDogeBsc

Telegram

https://t.me/OlympicDogeEN

AUDIT OVERVIEW

62

Security Score
HIGH RISK
Audit FAIL

93 **Static Scan**
Automatic scanning for common vulnerabilities

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

2	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) public onlyOwner {
        _isExcludedFromFees[account] = excluded;
}
```

🟡 Contract owner can exclude/include wallet from tx and wallet limitations

```
function excludeFromWalletLimit(address account, bool excluded) public onlyOwner {
        _isExcludedMaxWalletAmount[account] = excluded;
}

function excludeFromMaxTransaction(address updAds, bool isEx) public onlyOwner {
        _isExcludedMaxTransactionAmount[updAds] = isEx;
}
```

🔴 Contract owner is required to call the swapTrading() function in order to enable trading. Once trading is enabled, it can’t be disabled.

Please note that any wallet excluded from fees/taxes can still engage in trading even if trading is disabled.

```
function swapTrading() external onlyOwner {
        isTrading = true;
        swapEnabled = true;
        taxTill = block.number + 2;
}

trasfer function line 880
.
.
.
if (!isTrading) {
                require(_isExcludedFromFees[sender] || _isExcludedFromFees[recipient], "Trading is not active.");
          }
.
.
```

🔴 Contract owner can change buy fees up to 99%, sell fees up to 99%

```
function updateFees(uint256 _marketingFeeBuy, uint256 _liquidityFeeBuy,uint256 _contestAIFeeBuy,uint256
_marketingFeeSell, uint256 _liquidityFeeSell,uint256 _contestAIFeeSell) external onlyOwner{
    _fees.buyMarketingFee = _marketingFeeBuy;
    _fees.buyLiquidityFee = _liquidityFeeBuy;
    _fees.buyContestAIFee = _contestAIFeeBuy;
    _fees.buyTotalFees = _fees.buyMarketingFee + _fees.buyLiquidityFee + _fees.buyContestAIFee;

    _fees.sellMarketingFee = _marketingFeeSell;
    _fees.sellLiquidityFee = _liquidityFeeSell;
    _fees.sellContestAIFee = _contestAIFeeSell;
    _fees.sellTotalFees = _fees.sellMarketingFee + _fees.sellLiquidityFee + _fees.sellContestAIFee;
    require(_fees.buyTotalFees <= 99, "Must keep fees at 99% or less");
    require(_fees.sellTotalFees <= 99, "Must keep fees at 99% or less");

}
```

🟢 Contract owner can change marketingWallet and contestAIWallet addresses

Default values:

marketingWallet : 0x7DF8af89cA6c34e817838be21Fc5e91CD8A83C06

contestAIWallet : 0x1d070aCab995B7EA1A0f8AE3140F5BC38A4AB322

```
function setWallets(address _marketingWallet,address _contestAIWallet) external onlyOwner{
    marketingWallet = _marketingWallet;
    contestAIWallet = _contestAIWallet;
}
```

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

```
function updateMaxTxnAmount(uint256 newMaxBuy, uint256 newMaxSell) external onlyOwner {
    require(((totalSupply() * newMaxBuy) / 1000) >= (totalSupply() / 100), "Cannot set maxTransaction-
Amounts lower than 1%");
    require(((totalSupply() * newMaxSell) / 1000) >= (totalSupply() / 100), "Cannot set maxTransaction-
Amounts lower than 1%");
    maxBuyAmount = (totalSupply() * newMaxBuy) / 1000;
    maxSellAmount = (totalSupply() * newMaxSell) / 1000;
}


function updateMaxWalletAmount(uint256 newPercentage) external onlyOwner {
    require(((totalSupply() * newPercentage) / 1000) >= (totalSupply() / 100), "Cannot set maxWallet lower
than 1%");
    maxWalletAmount = (totalSupply() * newPercentage) / 1000;
}
```

Contract owner can change swap settings

```
function toggleSwapEnabled(bool enabled) external onlyOwner(){
        swapEnabled = enabled;
}

function updateThresholdSwapAmount(uint256 newAmount) external onlyOwner returns(bool){
        thresholdSwapAmount = newAmount;
        return true;
}
```

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;
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
        emit OwnershipTransferred(_owner, address(0));
        _owner = 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 2 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	10%
Sell fees:	10%
Max TX:	Not public
Max Sell:	Not public

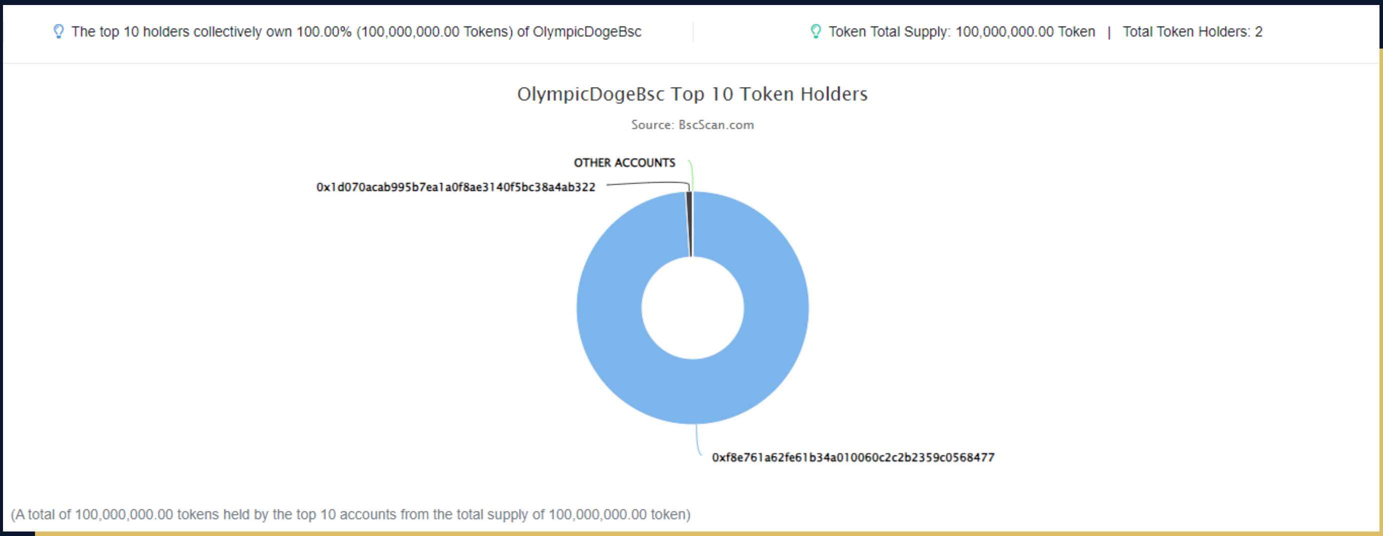
Honeypot Risk

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

OLYMPICDOGE TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



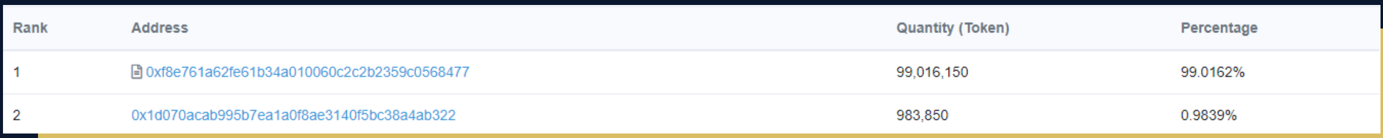
The top 10 holders collectively own 100.00% (100,000,000.00 Tokens) of OlympicDogeBsc

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

OlympicDogeBsc Top 10 Token Holders

Source: BscScan.com

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



Rank	Address	Quantity (Token)	Percentage
1	0xf8e761a62fe61b34a010060c2c2b2359c0568477	99,016,150	99.0162%
2	0x1d070acab995b7ea1a0f8ae3140f5bc38a4ab322	983,850	0.9839%

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.