



# SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



**Your Everyday Money**  
**\$YEM**

**05/03/2024**

# TOKEN OVERVIEW

---

## Fees

- Buy fees: 0%
- Sell fees: 0%

## Fees privileges

- Can change buy fees up to 14% and sell fees up to 14%

## Ownership

- Owned

## Minting

- No mint function

## Max Tx Amount / Max Wallet Amount

- Can change max wallet amount (without threshold)

## Blacklist

- Blacklist function not detected

## Other privileges

- Can exclude / include from fees
  - Contract owner has the ability to update the marketing wallet without requiring any additional checks
-

# TABLE OF CONTENTS

1

DISCLAIMER

2

INTRODUCTION

3

WEBSITE + SOCIALS

4-5

AUDIT OVERVIEW

6-8

OWNER PRIVILEGES

9

CONCLUSION AND ANALYSIS

10

TOKEN DETAILS

11

YEM ANALYTICS &  
TOP 10 TOKEN HOLDERS

12

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 **Your Everyday Money** (Customer) to conduct a Smart Contract Code Review and Security Analysis.

**0x8048aD07f54B383Ce6dFc144cD00741EC2b2Fc82**

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 **05/03/2024**



# WEBSITE DIAGNOSTIC

<http://www.yembinance.com>



0-49



50-89



90-100



Performance



Accessibility



Best  
Practices



SEO



Progressive  
Web App

## Socials



Twitter

N/A



Telegram

<https://t.me/yembinancechannelsafezone>

# 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 marketing wallet address

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

Current value: `0x906E08ddB7e5260A22b16E56AEbc59B7add91FE4`

```
function setMarketingWallet(address marketingWallet) external onlyOwner {
    _marketingWallet = marketingWallet;
}
```

## ● Contract owner can change max wallet amount limitation (without threshold)

Note that setting the value too low may prevent users from making purchase transactions

```
function setMaxWalletTokens(uint256 _maxToken) external onlyOwner {
    maxWalletToken = _maxToken;
}
```

## ● Contract owner can exclude/include wallet from tax

```
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 not 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 tax

```
function setExcludedFromFee(address account, bool e) external onlyOwner {
    _isExcludedFromFee[account] = e;
}
```

## ● Contract owner can change swap settings

```
function setSwapAndLiquifyEnabled(bool e) public onlyOwner {
    _swapAndLiquifyEnabled = e;
    emit SwapAndLiquifyEnabledUpdated(e);
}
```

## ● Contract owner can change fees up to 14%

```
function setTaxFeePercent(uint256 taxFee) external onlyOwner {
    require(taxFee <= 4, "Holder Reflection cannot exceed 4%");
    _taxFee = taxFee;
}

function setLiquidityFeePercent(uint256 liquidityFee) external onlyOwner {
    require(liquidityFee <= 10, "Liquidity Fee cannot exceed 10%");
    _liquidityFee = liquidityFee;
}
```

## ● 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:	0%
Sell fees:	0%
Max TX:	100,000,000
Max Wallet:	100,000,000

## Honeypot Risk

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

## Rug Pull Risk

Liquidity:	N/A
Holders:	100% unlocked tokens



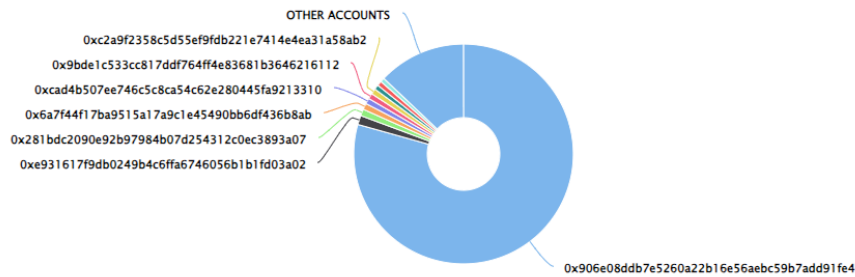
# YEM TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 87.07% (4,353,724,498.00 Tokens) of Your Everyday Money

Token Total Supply: 5,000,000,000.00 Token | Total Token Holders: 133

## Your Everyday Money Top 10 Token Holders

Source: BscScan.com



(A total of 4,353,724,498.00 tokens held by the top 10 accounts from the total supply of 5,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	<a href="#">0x906E08dd...7add91FE4</a>	3,967,094,498	79.3419%
2	<a href="#">0xE931617f...b1fD03A02</a>	67,000,000	1.3400%
3	<a href="#">0x281bDC20...EC3893a07</a>	50,000,000	1.0000%
4	<a href="#">0x6a7F44F1...dF436B8Ab</a>	45,000,000	0.9000%
5	<a href="#">0xCAD4B507...fa9213310</a>	42,000,000	0.8400%
6	<a href="#">0x9BDe1c53...646216112</a>	41,000,000	0.8200%
7	<a href="#">0xc2A9f235...A31a58aB2</a>	40,000,000	0.8000%
8	<a href="#">0xdBE77efe...A2F0d417c</a>	35,000,000	0.7000%
9	<a href="#">0xe225fA09...f8D5Bb1f8</a>	35,000,000	0.7000%
10	<a href="#">0x24D0165c...A8B473da8</a>	31,630,000	0.6326%

# 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.

