



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



UFOonaut Inu
\$UFI

12/03/2024



TOKEN OVERVIEW

Fees

- Buy fees: 5%
- Sell fees: 5%

Fees privileges

- Can change buy fees up to 5% and sell fees up to 5%

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can't change max tx amount or max wallet amount

Blacklist

- Blacklist function not detected

Other privileges

- Can exclude / include from fees
 - Contract owner has to call launch function to enable trade or adding LP
-

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

UFI 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 **UFOonaut Inu** (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x36aa402dcd144dc7ec5f0bee9b83047292e608b9

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 **12/03/2024**



WEBSITE DIAGNOSTIC

<https://www.ufonautinu.com/>



0-49



50-89



90-100



Performance



Accessibility



Best
Practices



SEO



Progressive
Web App

Socials



Twitter

<https://twitter.com/UFOonautInu>



Telegram

<https://t.me/UFOonautInuBsc>

AUDIT OVERVIEW



Security Score



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 an address from transactions
- Contract owner can exclude/include wallet(s) from tax

```
function excludeFromFees(address account, bool excluded) public onlyOwner {
    require(!_isExcludedFromFees[account] != excluded, "Account is already the value of 'excluded'");
    _isExcludedFromFees[account] = excluded;

    emit ExcludeFromFees(account, excluded);
}

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

- Contract owner can change swap settings (without threshold)

```
function SetSwapSeconds(uint256 newSeconds) external onlyOwner {
    _intervalSecondsForSwap = newSeconds * 1 seconds;
}

function setminimumTokensBeforeSwap (uint256 _new) public onlyOwner{
    minimumTokensBeforeSwap = _new;
}
```

- Contract owner has to call **launch** function to enable trade or adding LP

Please be aware that there is a 100-block delay before trading is enabled. The owner has the ability to set the maximum delay up to 200 blocks, as determined by the botTimer value.

Average block time on BSC Network is 3 seconds

```
function launch() public onlyOwner {
    launchedAt = block.number;
    _hasLiqBeenAdded = true;
}

_transferFrom function line 1030

if(!_hasLiqBeenAdded) {
    if(!_liquidityHolders[from] && to == uniswapV2Pair) {
        ...
    } else {
        if(launchedAt > 0
            ...
        }
    }
}
```

● Contract owner can change buy fees up to 5% and sell fees up to 5%

```
function setBuySellTax(uint256 _buyTax , uint256 _sellTax) external onlyOwner {  
    require(_buyTax <= 5, "Max is 5%!");  
    require(_sellTax <= 5, "Max is 5%!");  
    buyTax = _buyTax;  
    sellTax = _sellTax;  
}
```

● Contract owner can change fee distribution percentage

Current values:

liquidityFee: 2 uint256 (20%)

buybackFee: 8 uint256 (80%)

```
function setLiquidityFee(uint256 value) external onlyOwner{  
    liquidityFee = value;  
    totalFees = (liquidityFee).add(buybackFee);  
}  
  
function setBuybackFee(uint256 value) external onlyOwner{  
    buybackFee = value;  
    totalFees = (liquidityFee).add(buybackFee);  
}
```

● Contract owner can change `_buybackWalletAddress` address

Current value:

`_buybackWalletAddress` : 0xF24dab11a706394B912E6bCEfc9C6A361bb7D7Aa

```
function setBuybackWallet(address payable wallet) external onlyOwner{  
    _buybackWalletAddress = wallet;  
}
```

● Contract owner has the authority to modify the time delay until trading is enabled (with threshold)

Current value:

botTimer: 100 uint256

```
function botTimer(uint256 _timer) public onlyOwner{  
    require(_timer <= 200, "Bot timer can not be higher than 200 blocks ~ 10 minutes");  
    botTimer = _timer;  
}
```

● Contract owner can enable/disable snipe implementation

```
function changeSnipe(bool _snipe) external onlyOwner {  
    sniperProtection = _snipe;  
}
```

● Contract owner has ability to retrieve stuck BNB token held by the contract

```
function Sweep() external onlyOwner {
    uint256 balance = address(this).balance;
    payable(owner()).transfer(balance);
}
```

● Contract owner has ability to retrieve any token held by the contract

Native tokens not excluded

```
function transferContractToken(address _token, address _to, uint256 _quant) public onlyOwner returns(bool _sent){
    _sent = IERC20(_token).transfer(_to, _quant);
}
```

● 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 1 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees: 5%

Sell fees: 5%

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: Clean



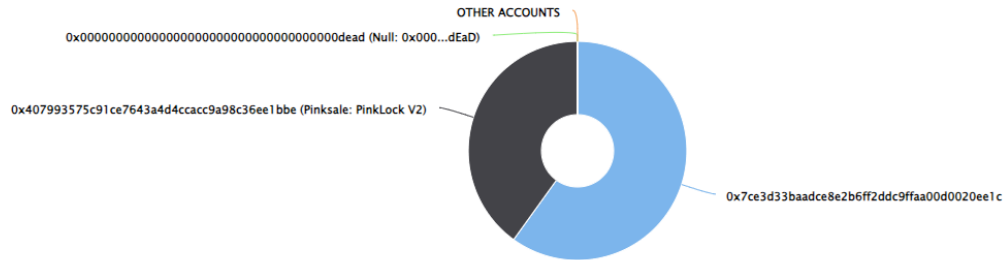
UFI TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

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

Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 3

UFOonaut Inu Top 10 Token Holders

Source: BscScan.com



(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	0x7Ce3d33b...D0020EE1c	599,550,000	59.9550%
2	Pinksale: PinkLock V2	400,000,000	40.0000%
3	Null: 0x000...dEaD	450,000	0.0450%

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.

