freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Botrix AI

$BTIX

07/04/2023

TOKEN OVERVIEW

Fees

• Buy fees:	3%
• Sell fees:	5%

Fees privileges

• Can change buy fees up to 7% and sell fees up to 7%

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

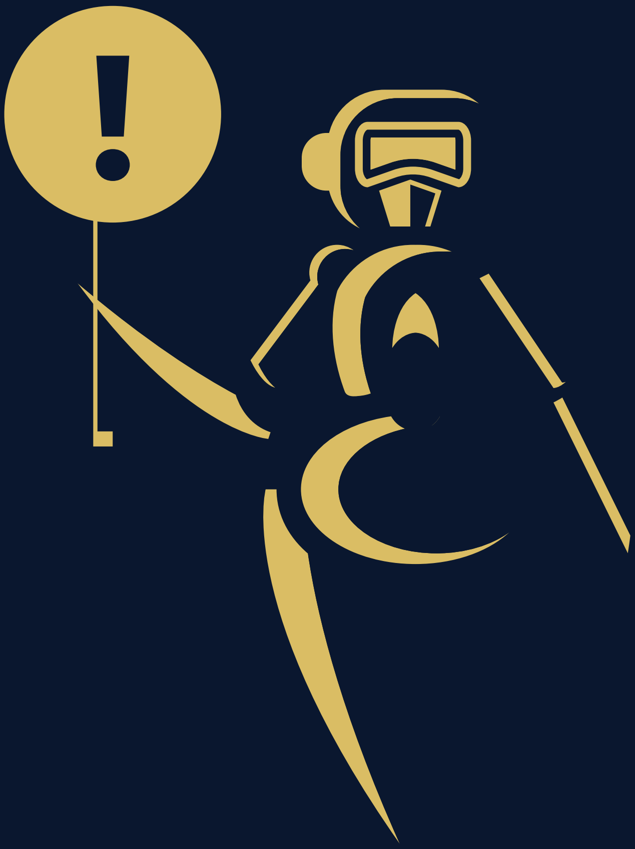
TABLE OF CONTENTS

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

0xBce4CE83005C969B812eD927D7bc4345cfA18f5B

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 07/04/2023

WEBSITE DIAGNOSTIC

https://botrix.io/

🟥	🟨	🟩
0-49	50-89	90-100

94	92	97	92	NA
Performance	Accessibility	Best Practices	SEO	Progressive Web App

Socials

Twitter

https://twitter.com/botrixai

Telegram

https://t.me/BotrixAI

AUDIT OVERVIEW

94

Security Score

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

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

0	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(s) from tax

```
function updateExemptFee(address _address, bool state) external onlyOwner {
   exemptFee[_address] = state;
}

function bulkExemptFee(address[] memory accounts, bool state) external onlyOwner {
   for (uint256 i = 0; i < accounts.length; i++) {
     exemptFee[accounts[i]] = state;
   }
}
```

🟢 Contract owner can change swap settings

```
function updateLiquidityProvide(bool state) external onlyOwner {
   // Update liquidity providing state
   providingLiquidity = state;
}

function updateLiquidityTreshhold(uint256 newAmount) external onlyOwner {
   // Update the treshhold (with limitations)
   require(
     newAmount <= totalSupply() / 100,
     'Swap threshold amount should be lower or equal to 1% of tokens'
   );
   tokenLiquidityThreshold = newAmount * 10 ** decimals();
}
```

🟡 Contract owner has to call enableTrading function to enable trade

(once enabled, can't be disabled)

Current value: tradingEnabled: false;

```
function enableTrading() external onlyOwner {
   require(!tradingEnabled, 'Cannot re-enable trading');
   tradingEnabled = true;
   providingLiquidity = true;
   genesisBlock = block.number;
}
```

● Contract owner can withdraw tokens from smart contract

(BTIX tokens excluded)

```
function rescueBNB(uint256 weiAmount) external onlyOwner {
    payable(owner()).transfer(weiAmount);
}

function rescueBEP20(address tokenAdd, uint256 amount) external onlyOwner {
    require(
      tokenAdd != address(this),
      "Owner can't claim contract's balance of its own tokens"
    );
    IBEP20(tokenAdd).transfer(owner(), amount);
}
```

● Contract owner can change buy fees up to 7% and sell fees up to 7%

```
function setBuyTaxes(
    uint256 _marketing,
    uint256 _liquidity,
    uint256 _development
  ) external onlyOwner {
    taxes = Taxes(_marketing, _liquidity, _development);
    require(
      (_marketing + _liquidity + _development) <= 7,
      'Must keep fees at 7% or less'
    );
}

function setSellTaxes(
    uint256 _marketing,
    uint256 _liquidity,
    uint256 _development
  ) external onlyOwner {
    sellTaxes = Taxes(_marketing, _liquidity, _development);
    require(
      (_marketing + _liquidity + _development) <= 7,
      'Must keep fees at 7% or less'
    );
}
```

● Contract owner can change _marketingWallet and _developmentWallet addresses

```
function updateMarketingWallet(address newWallet) external onlyOwner {
    require(newWallet != address(0), 'Fee Address cannot be zero address');
    _marketingWallet = newWallet;
}

function updateDevelopmentWallet(address newWallet) external onlyOwner {
    require(newWallet != address(0), 'Fee Address cannot be zero address');
    _developmentWallet = newWallet;
}
```

● Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(newOwner != address(0), 'Ownable: new owner is the zero address');
    _setOwner(newOwner);
}

function _setOwner(address newOwner) private {
    address oldOwner = _owner;
    _owner = newOwner;
    emit OwnershipTransferred(oldOwner, newOwner);
}
```

● Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    _setOwner(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 no HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	3%
Sell fees:	5%
Max TX:	N/A
Max Sell:	N/A

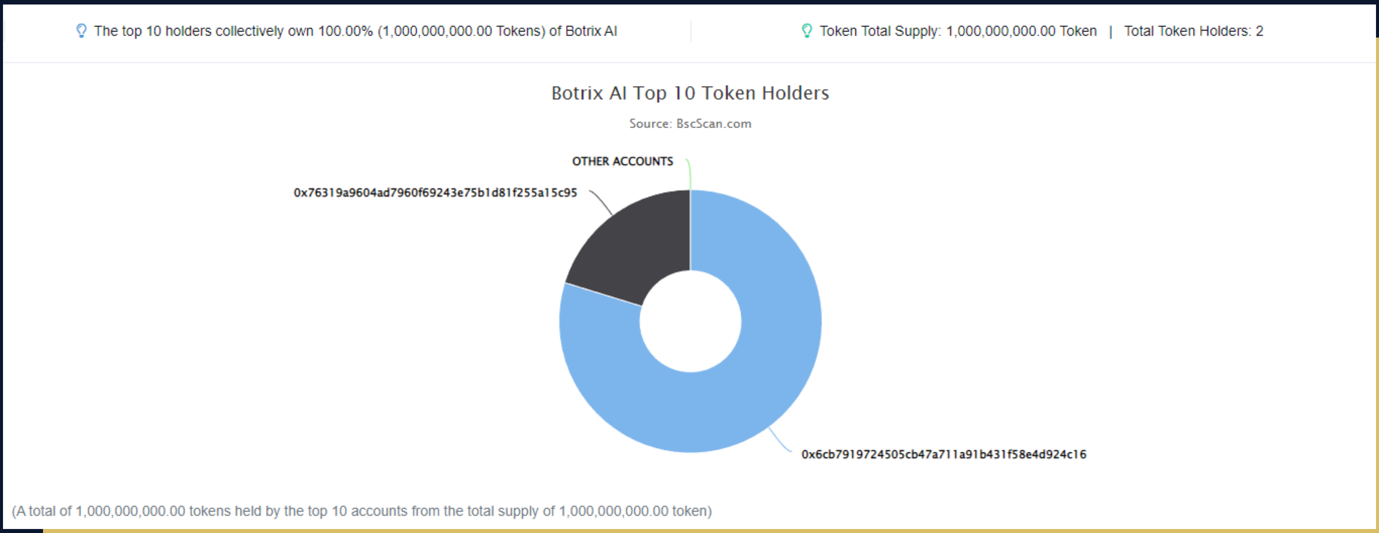
Honeypot Risk

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:	79,74% unlocked tokens

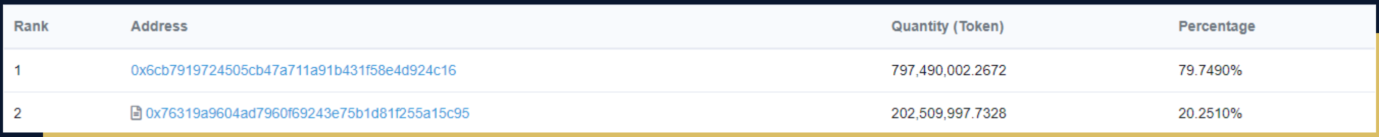
BTIX TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



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

Token Total Supply: 1,000,000,000.00 Token | Total Token Holders: 2

(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	0x6cb7919724505cb47a711a91b431f58e4d924c16	797,490,002.2672	79.7490%
2	0x76319a9604ad7960f69243e75b1d81f255a15c95	202,509,997.7328	20.2510%

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.