freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Ai Bonk

$AIBONK

09/03/2024

TOKEN OVERVIEW

Fees

• Buy fees:	1%
• Sell fees:	3%

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 change max tx amount and max wallet amount (without threshold)

Blacklist

• Blacklist function detected

Other privileges

• Can exclude / include from fees

• Can burn tokens

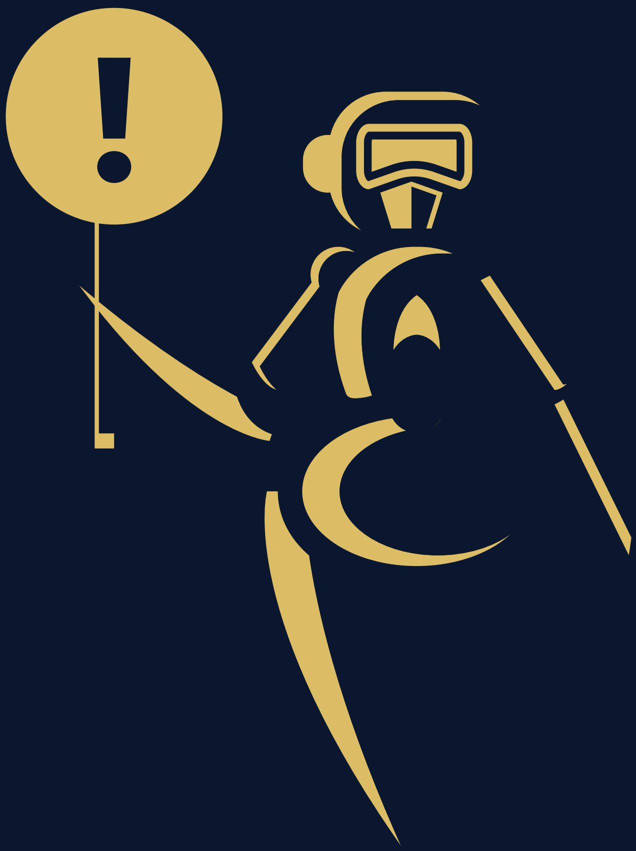
TABLE OF CONTENTS

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

0x3815810788686A9a53730199AFA392D81312f8c7

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 09/03/2024

WEBSITE DIAGNOSTIC

https://aibonk.vip/

0-49	50-89	90-100

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

Socials

Twitter

https://x.com/albonk

Telegram

https://t.me/AIBONK_aibonk

AUDIT OVERVIEW

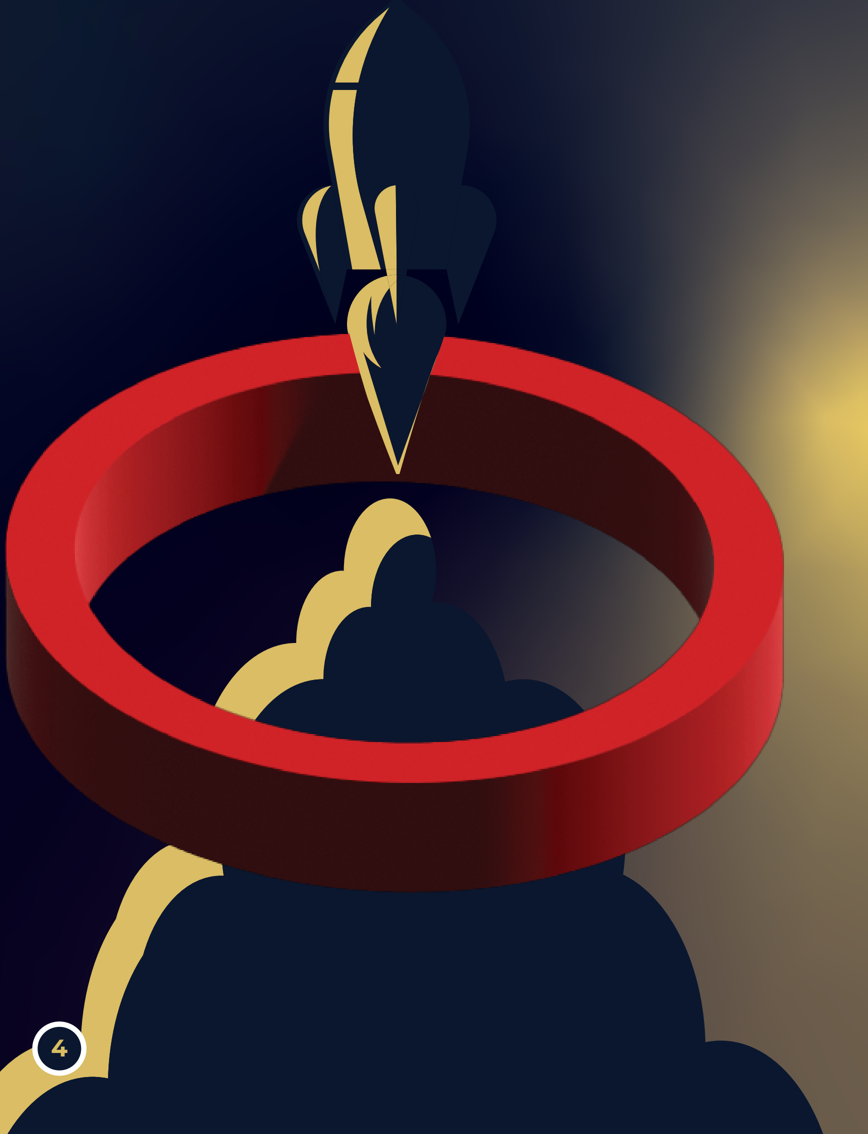
55

Security Score
HIGH RISK
Audit FAIL

93
Static Scan
Automatic scanning for common vulnerabilities

52
ERC Scan
Automatic checks for ERC's conformance

3	High
1	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	Low
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 exclude/include wallet from fees

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

      _isExcludedFromFee[account] = excluded;
}
```

🔴 Contract owner can exclude/include addresses from transactions

```
function blockAccount(address account) external onlyOwner {
      require(!_isBlocked[account], "TOKEN: Account is already blocked");
      _isBlocked[account] = true;
}

function unblockAccount(address account) external onlyOwner {
      require(_isBlocked[account], "TOKEN: Account is not blcoked");
      _isBlocked[account] = false;
}
```

🔴 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 setMaxWalletAmount(uint256 newValue) external onlyOwner {
      require(newValue != maxWalletAmount, "TOKEN: Cannot update maxWalletAmount to same value");

      maxWalletAmount = newValue;
}
```

🔴 Contract owner can change max tx amount limitation (without threshold)

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

```
function setMaxTransactionAmount(uint256 newValue) external onlyOwner {
      require(newValue != maxTxAmount, "TOKEN: Cannot update maxTxAmount to same value");

      maxTxAmount = newValue;
}
```

🟢 Tokens can be burned

```
function burn(uint256 amount) public virtual {
      _burn(msg.sender, amount);
}
```

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

```
function excludeFromMaxTransactionLimit(address account, bool excluded) external onlyOwner {
      require(
         _isExcludedFromMaxTransactionLimit[account] != excluded,
         "TOKEN: Account is already the value of 'excluded'"
      );
      _isExcludedFromMaxTransactionLimit[account] = excluded;
}

function excludeFromMaxWalletLimit(address account, bool excluded) external onlyOwner {
      require(
         _isExcludedFromMaxWalletLimit[account] != excluded,
         "TOKEN: Account is already the value of 'excluded'"
      );
      _isExcludedFromMaxWalletLimit[account] = excluded;
}
```

● Contract owner can change swap settings (without threshold)

```
function setMinimumTokensBeforeSwap(uint256 newValue) external onlyOwner {
      require(newValue != minimumTokensBeforeSwap, "TOKEN: Cannot update minimumTokensBeforeSwap
to same value");

      minimumTokensBeforeSwap = newValue;
}
```

● Contract owner can change buy fees up to 14% and sell fees up to 14%

```
function setFeesData(FeeDataStorage calldata taxData) external onlyOwner {
      require(
         (taxData.liquidityFeeOnBuy + taxData.operationsFeeOnBuy + taxData.burnFeeOnBuy) <= 25,
         "TOKEN: Tax exceeds maximum value of 30%"
      );
      require(
         (taxData.liquidityFeeOnSell + taxData.operationsFeeOnSell + taxData.burnFeeOnSell) <= 25,
         "TOKEN: Tax exceeds maximum value of 30%"
      );

      baseFeeData = taxData;
}
```

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

```
function claimETHOverflow() external onlyOwner {
      uint256 amount = address(this).balance;

      (bool success, ) = address(owner()).call{ value: amount }("");

      require(success);
}
```

Contract owner can change liquidityWallet and operationsWallet addresses

```
function setWallets(address newLiquidityWallet, address newOperationsWallet) external onlyOwner {
        if (liquidityWallet != newLiquidityWallet) {
            require(newLiquidityWallet != address(0), "TOKEN: The liquidityWallet cannot be 0");
            liquidityWallet = newLiquidityWallet;
        }

        if (operationsWallet != newOperationsWallet) {
            require(newOperationsWallet != address(0), "TOKEN: The operationsWallet cannot be 0");
            operationsWallet = newOperationsWallet;
        }
}
```

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));
}
```

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 3 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	1%
Sell fees:	3%
Max TX:	240,000,000
Max Wallet:	240,000,000

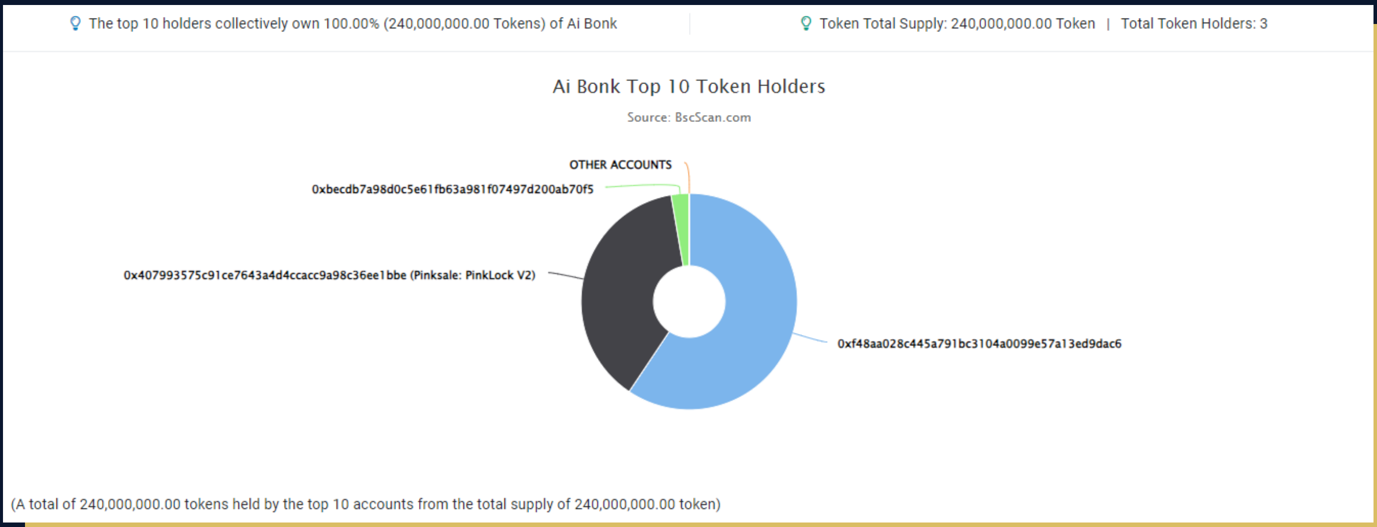
Honeypot Risk

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

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

AIBONK TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS



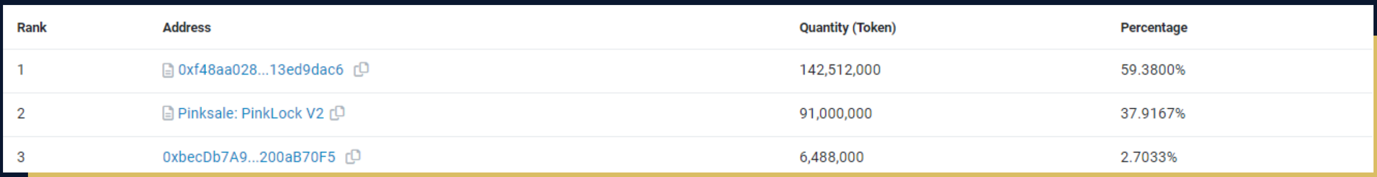
The top 10 holders collectively own 100.00% (240,000,000.00 Tokens) of Ai Bonk

Token Total Supply: 240,000,000.00 Token | Total Token Holders: 3

Ai Bonk Top 10 Token Holders

Source: BscScan.com

(A total of 240,000,000.00 tokens held by the top 10 accounts from the total supply of 240,000,000.00 token)



Rank	Address	Quantity (Token)	Percentage
1	0xf48aa028...13ed9dac6	142,512,000	59.3800%
2	Pinksale: PinkLock V2	91,000,000	37.9167%
3	0xbecDb7A9...200aB70F5	6,488,000	2.7033%

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.