



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



VisionScanAi
\$VSAI

10/03/2023

TOKEN OVERVIEW

Fees

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

Fees privileges

- Can't change fees

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

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

Blacklist

- Blacklist function not detected

Other privileges

- Can enable / disable fees
-

TABLE OF CONTENTS

1

DISCLAIMER

2

INTRODUCTION

3

WEBSITE + SOCIALS

4-5

AUDIT OVERVIEW

6-7

OWNER PRIVILEGES

8

CONCLUSION AND ANALYSIS

9

TOKEN DETAILS

10

VSAI TOKEN ANALYTICS &
TOP 10 TOKEN HOLDERS

11

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

0xC76ebf216C944ea5bd3160b6100F6CdFab01CaeE

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 **10/03/2023**



WEBSITE DIAGNOSTIC

N/A



0-49



50-89



90-100



Performance



Accessibility



Best
Practices



SEO



Progressive
Web App

Socials



Twitter

<https://twitter.com/visionscanai>



Telegram

<https://t.me/visionscanai>

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	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 include / exclude 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 {
    providingLiquidity = state;
}

function updateLiquidityTreshhold(uint256 new_amount) external onlyOwner {
    require(new_amount <= 1e5, "Swap threshold amount should be lower or equal to 1% of tokens");
    tokenLiquidityThreshold = new_amount * 10**decimals();
}
```

- Contract owner can change marketingWallet address

Current value:

marketingWallet: 0xede6f5600681cd9349865a2011e5ddc4319909a1

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

- Contract owner has to call EnableTrading function to enable trade once enabled, can't be disabled

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

● Contract owner can withdraw tokens from smart contract

except own tokens

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

function rescueBSC20(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 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));
}
```



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: 0%

Sell fees: 0%

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



VSAI TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

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

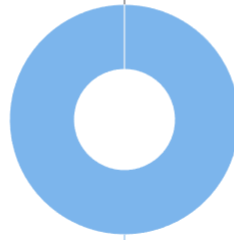
Token Total Supply: 1,000,000,000.00 Token

Total Token Holders: 1

VisionScanAi Top 10 Token Holders


Source: BscScan.com

OTHER ACCOUNTS



0xede6f5600681cd9349865a2011e5ddc4319909a1

(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	0x9d1d4f83b340329aaf18db636d90676b3fcc39d9	555,920,000	55.5920%
2	 Pinksale: PinkLock V2	444,080,000	44.4080%

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.

