freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

AGRITECH

$AGT

14/03/2023

TOKEN OVERVIEW

Fees

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

Fees privileges

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

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change / set max tx amount or max wallet amount (with threshold)

Blacklist

- Blacklist function not detected

Other privileges

- Can enable / disable fees

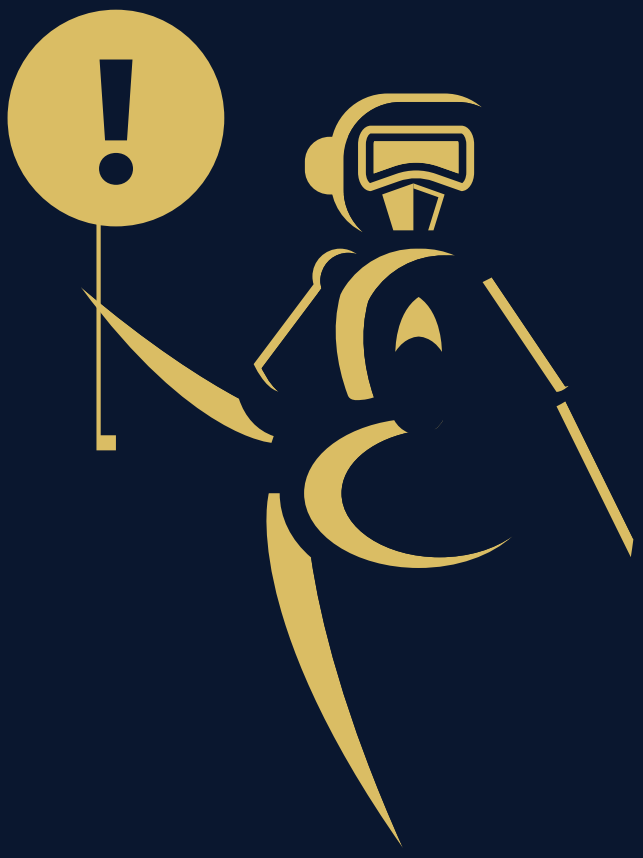
TABLE OF CONTENTS

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

0xabc10C60Abfd6a001E83BC7172d478DfF2F70a9c

Network: Ethereum (ETH)

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 14/03/2023

WEBSITE DIAGNOSTIC

http://agri-tech.io/


0-49


50-89


90-100

Performance

Accessibility

Best Practices

SEO

Progressive Web App

Socials


Twitter

https://twitter.com/AgritechAI


Telegram

https://t.me/Agritechchannel

AUDIT OVERVIEW

91

Security Score

94

Static Scan
Automatic scanning for common vulnerabilities

86

ERC Scan
Automatic checks for ERC's conformance

0 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	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 from tax

```
function setExcludedFromFees(address account, bool enabled) public onlyOwner {
    _isExcludedFromFees[account] = enabled;
}
```

- Contract owner can change _marketingWallet and _developmentWallet addresses

Default values:

_marketingWallet: 0x300651Bc7b5CF87f40AE738Efe007f3737Bdcce8

_developmentWallet: 0x65e3baF7631DF8364FE4CF09f257E03C95D964D2

```
function setWallets(address payable marketingWallet, address payable developmentWallet) external
onlyOwner {
    _marketingWallet = payable(marketingWallet);
    _developmentWallet = payable(developmentWallet);
}
```

- Contract owner can change max tx amount and max wallet (with threshold)

```
function setMaxTxPercent(uint256 percent, uint256 divisor) external onlyOwner {
    uint256 check = (_tTotal * percent) / divisor;
    require(check >= (_tTotal / 1000), "Must be above 0.1% of total supply.");
    _maxTxAmount = check;
    maxTxAmountUI = (startingSupply * percent) / divisor;
}

function setMaxWalletSize(uint256 percent, uint256 divisor) external onlyOwner {
    uint256 check = (_tTotal * percent) / divisor;
    require(check >= (_tTotal / 1000), "Must be above 0.1% of total supply.");
    _maxWalletSize = check;
    maxWalletSizeUI = (startingSupply * percent) / divisor;
}
```

- Contract owner can change swap settings

```
function setSwapSettings(uint256 thresholdPercent, uint256 thresholdDivisor, uint256 amountPercent,
uint256 amountDivisor) external onlyOwner {
    swapThreshold = (_tTotal * thresholdPercent) / thresholdDivisor;
    swapAmount = (_tTotal * amountPercent) / amountDivisor;
}
```

Contract owner can change transfer tax up to 5%, buy tax up to 5% and sell tax up to 5%. Contract owner can change tax ratios.

```
uint256 constant public maxBuyTaxes = 500;
uint256 constant public maxSellTaxes = 500;
uint256 constant public maxTransferTaxes = 500;

uint256 public _liquidityRatio = 1;
uint256 public _marketingRatio = 2;
uint256 public _developmentRatio = 2;
```

```
function setTaxes(uint256 buyFee, uint256 sellFee, uint256 transferFee) external onlyOwner {
    require(buyFee <= maxBuyTaxes
        && sellFee <= maxSellTaxes
        && transferFee <= maxTransferTaxes,
        "Cannot exceed maximums.");
    _buyFee = buyFee;
    _sellFee = sellFee;
    _transferFee = transferFee;
}

function setRatios(uint256 liquidity, uint256 marketing, uint256 development) external onlyOwner {
    require (liquidity + marketing == 100, "Must add up to 100%");
    _liquidityRatio = liquidity;
    _marketingRatio = marketing;
    _developmentRatio = development;
}
```

Contract owner can transfer ownership

```
function transferOwner(address newOwner) external onlyOwner() {
    require(newOwner != address(0), "Call renounceOwnership to transfer owner to the zero address.");
    require(newOwner != DEAD, "Call renounceOwnership to transfer owner to the zero address.");
    setExcludedFromFees(_owner, false);
    setExcludedFromFees(newOwner, true);

    if (_marketingWallet == payable(_owner))
       _marketingWallet = payable(newOwner);

    _allowances[_owner][newOwner] = balanceOf(_owner);
    if(balanceOf(_owner) > 0) {
       _transfer(_owner, newOwner, balanceOf(_owner));
    }

    _owner = newOwner;
    emit OwnershipTransferred(_owner, newOwner);
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner() {
    setExcludedFromFees(_owner, false);
    _owner = address(0);
    emit OwnershipTransferred(_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 no HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	5%
Sell fees:	5%
Transfer fees:	5%
Max TX:	5,000,000
Max Sell:	N/A

Honeypot Risk

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

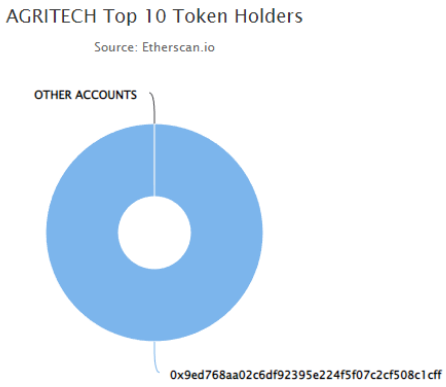
Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

AGT TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

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

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



(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	0x9eD768...508c1CfF	1,000,000,000	100.0000%

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.