freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

VEGA

$VEGA

22/11/2022

TOKEN OVERVIEW

Fees

- Buy fees: N/A
- Sell fees: N/A

Fees privileges

- Can change fees up to 100%

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change max tx amount and max wallet amount at a very low value

Blacklist

- Blacklist function detected

Other privileges

- Can exclude / include from fees
- Can burn tokens from a specific wallet

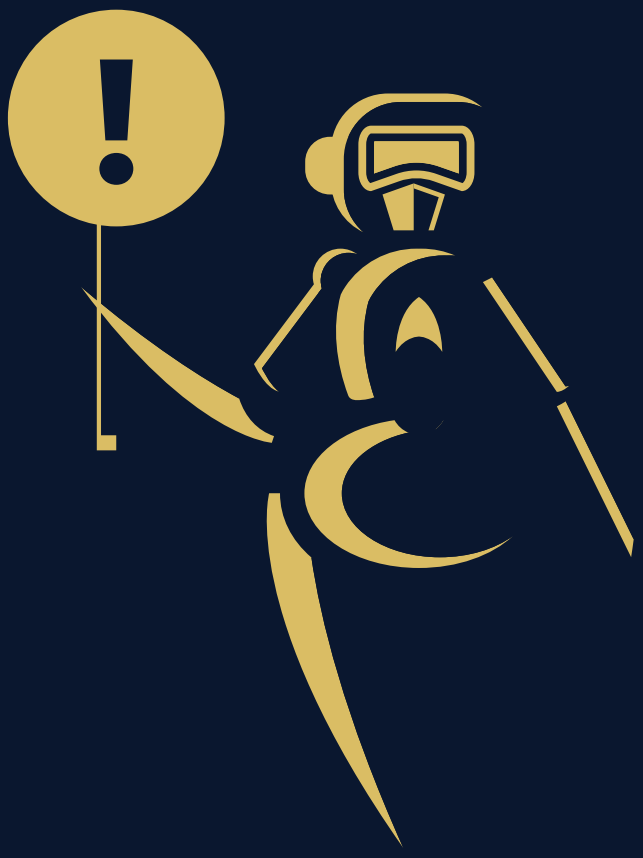
TABLE OF CONTENTS

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

0x2D7d34cD364FC5Ff3F53fa6D4D9aD017Ec8ef9A6

Network: Polygon (MATIC)

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 22/11/2022

AUDIT OVERVIEW

41

Security Score

HIGH RISK

93

Static Scan

Automatic scanning for common vulnerabilities

30

ERC Scan

Automatic checks for ERC's conformance

4 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

interface IAccessControl (Role lib included)

- Contract owner can exclude an address from transactions with grantRole function and 0x98db8a220cd0f09badce9f22d0ba7e93edb3d404448cc3560d391ab096ad16e9 argument
- Contract owner can exclude/include wallet(s) from tax

```
function excludeFromFee(address account) public onlyRole(MMER_ROLE) {
    super._excludeFromFee(account);
}

function includeInFee(address account) public onlyRole(MMER_ROLE) {
    return super._includeInFee(account);
}

function _excludeFromFee(address account) public virtual {
    _isExcludedFromFee[account] = true;
}

function _includeInFee(address account) public virtual {
    _isExcludedFromFee[account] = false;
}
```

- Contract owner can burn tokens from specific wallet

```
function burn(uint256 amount) public virtual {
    _burn(_msgSender(), amount);
}

function burnFrom(address account, uint256 amount) public virtual {
    _spendAllowance(account, _msgSender(), amount);
    _burn(account, amount);
}
```

- Contract owner can set max tx amount and max wallet amount at a a very low value

```
function setMaxTokenHolder(uint256 newMaxTokenHolder_) external onlyRole(MMER_ROLE) {
    require(newMaxTokenHolder_ > 0, "Fee amount must be greater than zero");
    _maxTokenHolder = newMaxTokenHolder_;
}

function setMaxTxAmount(uint256 maxTxAmount_) external onlyRole(MMER_ROLE) {
    require(maxTxAmount_ > 0, "Fee amount must be greater than zero");
    _maxTxAmount = maxTxAmount_;
}
```

Contract owner can change fees up to 100%

```
function setTaxBuy(uint256 fee_) external onlyRole(MMER_ROLE) {
    require(fee_ >= 0, "Fee amount must be greater than zero");
    _feeBuyPercent = fee_;
    emit UpdateTaxInfo("Buy Fee", _msgSender(), _feeBuyPercent);
}

function setTaxSell(uint256 fee_) external onlyRole(MMER_ROLE) {
    require(fee_ >= 0, "Fee amount must be greater than zero");
    _feeSellPercent = fee_;
    emit UpdateTaxInfo("Sell Fee", _msgSender(), _feeSellPercent);
}

function setTaxTransfer(uint256 fee_) external onlyRole(MMER_ROLE) {
    require(fee_ >= 0, "Fee amount must be greater than zero");
    _feeTransactionPercent = fee_;
    emit UpdateTaxInfo("Transaction Fee", _msgSender(), _feeTransactionPercent);
}

function setClaimTax(uint256 fee_) external onlyRole(MMER_ROLE) {
    require(fee_ >= 0, "Fee amount must be greater than zero");
    _feeClaimPercent = fee_;
    emit UpdateTaxInfo("ClaimReward Fee", _msgSender(), _feeClaimPercent);
}

function setBuyTicketTax(uint256 fee_) external onlyRole(MMER_ROLE) {
    require(fee_ >= 0, "Fee amount must be greater than zero");
    _feeBuyTicketPercent = fee_;
    emit UpdateTaxInfo("Buy Ticket Fee", _msgSender(), _feeBuyTicketPercent);
}

function setMarketingBuyTax(uint256 fee_) external onlyRole(MMER_ROLE) {
    require(fee_ >= 0, "Fee amount must be greater than zero");
    _walletMKTBuyPercent = fee_;
    emit UpdateTaxInfo("Marketing Buy Ticket Fee", _msgSender(), _walletMKTBuyPercent);
}

function setMarketingSellTax(uint256 fee_) external onlyRole(MMER_ROLE) {
    require(fee_ >= 0, "Fee amount must be greater than zero");
    _walletMKTSellPercent = fee_;
    emit UpdateTaxInfo("Marketing Claim Reward Fee", _msgSender(), _walletMKTSellPercent);
}
```

Contract owner can change _marketingAddr address

Current value:

_marketingAddr : 0x3ea0d8b81650918c6f571002bef7347826ac2eca

```
function setMarketingWallet(address mktAddr_) external onlyRole(DEFAULT_ADMIN_ROLE) {
    _marketingAddr = mktAddr_;
}
```

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 4 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	N/A
Sell fees:	N/A
Max TX:	300,000,000
Max Sell:	N/A

Honeypot Risk

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

Others

Liquidity:	N/A
Holders:	Clean

VEGA TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (100,000,000,000.00 Tokens) of VEGA

Token Total Supply: 100,000,000,000.00 Token | Total Token Holders: 1

(A total of 100,000,000,000.00 tokens held by the top 10 accounts from the total supply of 100,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0xa65c6a6bd8a2f510872457be7a8dafcece248596	100,000,000,000	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.