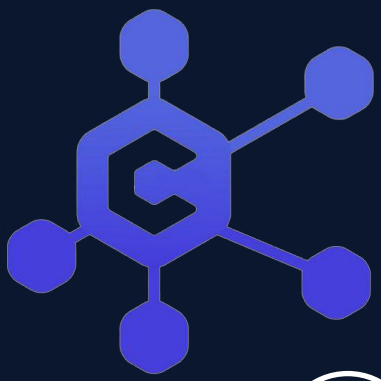
freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Cryptonomics

$Cryo

03/12/2022

TOKEN OVERVIEW

Fees

- Buy fees: 7%
- Sell fees: 7%

Fees privileges

- Can set 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 without threshold

Blacklist

- Blacklist function detected

Other privileges

- Can exclude / include from fees
- Can exclude / include from dividends
- Can exclude / include from tx limitations and wallet limitations

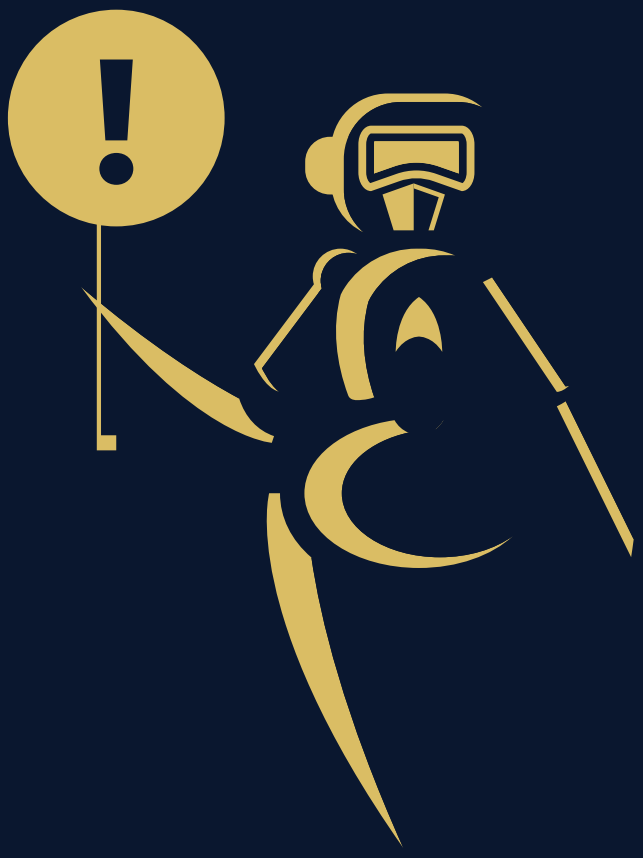
TABLE OF CONTENTS

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

0x31AEF016A16A669Bef148DFa6F1C892B7715620e

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 03/12/2022

AUDIT OVERVIEW

52

Security Score
HIGH RISK

93

Static Scan
Automatic scanning for common vulnerabilities

41

ERC Scan
Automatic checks for ERC's conformance

4 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 exclude an address from transactions

```
function updateIsBlacklisted(address account, bool state) external onlyOwner{
        _isBlacklisted[account] = state;
}

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

- Contract owner can exclude/include wallet(s) from tax

```
function excludeFromFee(address account) public onlyOwner {
        _isExcludedFromFee[account] = true;
}

function includeInFee(address account) public onlyOwner {
        _isExcludedFromFee[account] = false;
}

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

- Contract owner can exclude/include wallet from rewards

```
function excludeFromReward(address account) public onlyOwner() {
        require(!_isExcluded[account], "Account is already excluded");
        if(_rOwned[account] > 0) {
            _tOwned[account] = tokenFromReflection(_rOwned[account]);
        }
        _isExcluded[account] = true;
        _excluded.push(account);
}

function includeInReward(address account) external onlyOwner() {
        require(_isExcluded[account], "Account is not excluded");
        for (uint256 i = 0; i < _excluded.length; i++) {
            if (_excluded[i] == account) {
                _excluded[i] = _excluded[_excluded.length - 1];
                _tOwned[account] = 0;
                _isExcluded[account] = false;
                _excluded.pop();
                break;
            }
        }
}
```

Contract owner can change fees up to 100%

```
function setTaxes(uint256 _rfi, uint256 _development, uint256 _liquidity, uint256 _marketing) public
onlyOwner {
    taxes = Taxes(_rfi,_development,_liquidity,_marketing);
    emit FeesChanged();
}

function setSellTaxes(uint256 _rfi, uint256 _development, uint256 _liquidity, uint256 _marketing) public
onlyOwner {
    sellTaxes = Taxes(_rfi,_development,_liquidity,_marketing);
    emit FeesChanged();
}
```

Contract owner can change marketingWallet, developmentWallet addresses

Current values:

marketingWallet : 0x607c1fc72ff68638d989aa98681ea47500cabba6

developmentWallet : 0x607c1fc72ff68638d989aa98681ea47500cabba6

```
function updatemarketingWallet(address newWallet) external onlyOwner{
    marketingWallet = newWallet;
}

function updatedevelopmentWallet(address newWallet) external onlyOwner{
    developmentWallet = newWallet;
}
```

Contract owner can change swap settings

```
function updateSwapTokensAtAmount(uint256 amount) external onlyOwner{
    swapTokensAtAmount = amount * 10**_decimals;
}

function updateSwapEnabled(bool _enabled) external onlyOwner{
    swapEnabled = _enabled;
}
```

Contract owner can change max tx amount limitations and max wallet amount to 0 value

```
function updateMaxTxLimit(uint256 maxBuy, uint256 maxSell) external onlyOwner{
    maxBuyLimit = maxBuy * 10**decimals();
    maxSellLimit = maxSell * 10**decimals();
}

function updateMaxWalletlimit(uint256 amount) external onlyOwner{
    maxWalletLimit = amount * 10**decimals();
}
```

Contract owner can change cooldown between trades status and interval value

```
function updateCooldown(bool state, uint256 time) external onlyOwner{
    coolDownTime = time * 1 seconds;
    coolDownEnabled = state;
}
```

Contract owner can enable/disable trading

```
function setTradingStatus(bool state) external onlyOwner{
    tradingEnabled = state;
    swapEnabled = state;
    if(state == true && genesis_block == 0) genesis_block = block.number;
}
```

Contract owner can transfer tokens to specific wallet(s)

```
function airdropTokens(address[] memory accounts, uint256[] memory amounts) external onlyOwner{
    require(accounts.length == amounts.length, "Arrays must have same size");
    for(uint256 i = 0; i < accounts.length; i++){
        _tokenTransfer(msg.sender, accounts[i], amounts[i], false, false);
    }
}
```

Contract owner can withdraw stuck tokens from smart contract

```
function rescueBNB(uint256 weiAmount) external onlyOwner{
    require(address(this).balance >= weiAmount, "insufficient BNB balance");
    payable(msg.sender).transfer(weiAmount);
}

function rescueAnyBEP20Tokens(address _tokenAddr, address _to, uint _amount) public onlyOwner {
    IERC20(_tokenAddr).transfer(_to, _amount);
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    _setOwner(address(0));
}
```

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);
}
```

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:	7%
Sell fees:	7%
Max TX:	20,000
Max Sell:	20,000

Honeypot Risk

Ownership:	Owned
Blacklist:	Detected
Modify Max TX:	Detected
Modify Max Sell:	Detected
Disable Trading:	Detected

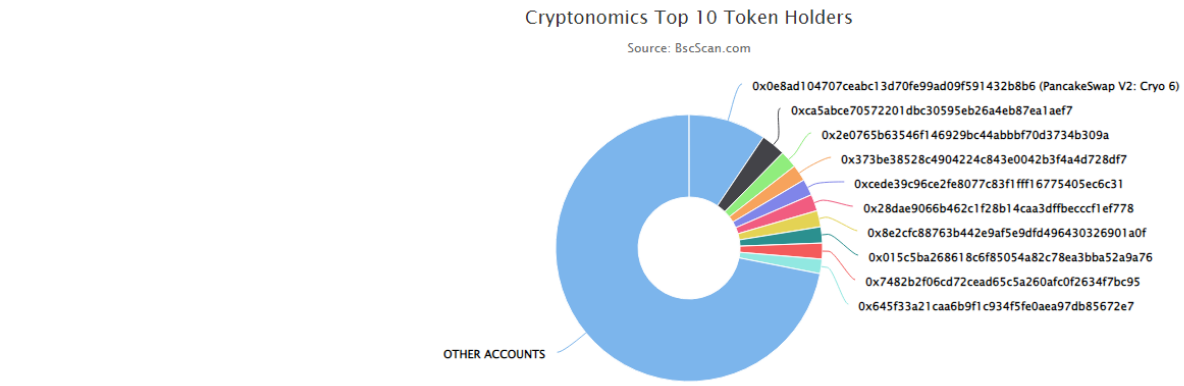
Others

Liquidity:	N/A
Holders:	Clean

CRYPTONOMICS TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 28.09% (280,936.65 Tokens) of Cryptonomics

Token Total Supply: 1,000,000.00 Token | Total Token Holders: 341



(A total of 280,936.65 tokens held by the top 10 accounts from the total supply of 1,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	PancakeSwap V2: Cryo 6	94,235.843137	9.4236%
2	0xca5abce70572201dbc30595eb26a4eb87ea1aef7	29,504.474113	2.9504%
3	0x2e0765b63546f146929bc44abbbf70d3734b309a	21,149.943696	2.1150%
4	0x373be38528c4904224c843e0042b3f4a4d728df7	19,980	1.9980%
5	0xcede39c96ce2fe8077c83f1fff16775405ec6c31	19,936	1.9936%
6	0x28dae9066b462c1f28b14caa3dffbecccf1ef778	19,918.104655	1.9918%
7	0x8e2cfc88763b442e9af5e9dfd496430326901a0f	19,750.288806	1.9750%
8	0x015c5ba268618c6f85054a82c78ea3bba52a9a76	19,560	1.9560%
9	0x7482b2f06cd72cead65c5a260afc0f2634f7bc95	19,300	1.9300%
10	0x645f33a21caa6b9f1c934f5fe0aea97db85672e7	17,602	1.7602%

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.