freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

CODEX

$CODEX

08/12/2023

TOKEN OVERVIEW

Fees

• **Buy fees:**   5%

• **Sell fees:**   5%

Fees privileges

• Can change 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 max tx amount and / or max wallet amount (with threshold)

Blacklist

• Blacklist function not detected

Other privileges

• Can exclude / include from fees

• Contract owner has to set true setTrading function to enable trade

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

0xcbe11cab348ac489ff2cbe4b19db853b2a568a1e

Network: Ethereum (Goerli Testnet)

This report presents the findings of the security assessment of
Customer's smart contract and its code review conducted on 08/12/2023

WEBSITE DIAGNOSTIC

https://codextoken.com/

0-49   50-89   90-100

94
Performance

92
Accessibility

94
Best
Practices

91
SEO

NA
Progressive
Web App

Socials

Twitter

https://twitter.com/codex_token

Telegram

https://t.me/codex_coin

AUDIT OVERVIEW

82

Security Score

98

Static Scan
Automatic scanning for common vulnerabilities

81

ERC Scan
Automatic checks for ERC's conformance

1 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	Low
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 exclude/include wallet(s) from tax

```
function excludeMultipleAccountsFromFees(address[] calldata accounts, bool excluded) public onlyOwner {
    for(uint256 i = 0; i < accounts.length; i++) {
        _isExcludedFromFee[accounts[i]] = excluded;
    }
}
```

- Contract owner has to set true setTrading function to enable trade

Please note that owner wallet retains the ability to engage in trading, even in situations where trading has been disabled. Trading once enabled, can't be disabled.

```
function setTrading(bool _tradingOpen) public onlyOwner {
    require(!tradingOpen, "Cannot reenable trading");
    tradingOpen = _tradingOpen;
}

_transferFrom function line 348
.
.
.
if (!tradingOpen) {
        require(from == owner(), "TOKEN: This account cannot send tokens until trading is enabled");
}
.
.
.
```

- Contract owner can change swap settings

```
function setMinSwapTokensThreshold(uint256 swapTokensAtAmount) public onlyOwner {
    _swapTokensAtAmount = swapTokensAtAmount;
}

function toggleSwap(bool _swapEnabled) public onlyOwner {
    swapEnabled = _swapEnabled;
}
```

● Contract owner can change buy fees up to 5% and sell fees up to 5%

```
function setFee(uint256 redisFeeOnBuy, uint256 redisFeeOnSell, uint256 taxFeeOnBuy, uint256 taxFeeOnSell)
public onlyOwner {
    require(redisFeeOnBuy >= 0 && redisFeeOnBuy <= 0, "Buy rewards must be between 0% and 0%");
    require(taxFeeOnBuy >= 0 && taxFeeOnBuy <= 5, "Buy tax must be between 0% and 25%");
    require(redisFeeOnSell >= 0 && redisFeeOnSell <= 0, "Sell rewards must be between 0% and 0%");
    require(taxFeeOnSell >= 0 && taxFeeOnSell <= 5, "Sell tax must be between 0% and 98%");

    _redisFeeOnBuy = redisFeeOnBuy;
    _redisFeeOnSell = redisFeeOnSell;
    _taxFeeOnBuy = taxFeeOnBuy;
    _taxFeeOnSell = taxFeeOnSell;
}
```

● Contract owner can change max tx amount limitation (with threshold)

```
function setMaxTxnAmount(uint256 maxTxAmount) public onlyOwner {
    require(maxTxAmount >= totalSupply().div(100), "value too low");
    _maxTxAmount = maxTxAmount;
}
```

● Contract owner can change max wallet amount limitation (with threshold)

```
function setMaxWalletSize(uint256 maxWalletSize) public onlyOwner {
    require(maxWalletSize >= totalSupply().div(100), "value too low");
    _maxWalletSize = maxWalletSize;
}
```

● Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {
    require(newOwner != address(0), "Ownable: new owner is the zero address");
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
}
```

● Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _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 1 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees: 5%

Sell fees: 5%

Max TX: 2,000,000

Max Sell: 2,000,000

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

CODEX TOKEN ANALYTICS
& TOP 10 TOKEN HOLDERS

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.