freshcoins

SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT

Error404

$ERROR404

17/02/2024

TOKEN OVERVIEW

Fees

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

Fees privileges

- Can’t change / set fees

Ownership

- Owned

Minting

- No mint function

Max Tx Amount / Max Wallet Amount

- Can change max tx amount (without threshold)

Blacklist

- Blacklist function not detected

Other privileges

- Can exclude / include from tx limitations
- Contract owner has the ability to change token name and token symbol

TABLE OF CONTENTS

1 DISCLAIMER

2 INTRODUCTION

3 WEBSITE + SOCIALS

4-5 AUDIT OVERVIEW

6-8 OWNER PRIVILEGES

9 CONCLUSION AND ANALYSIS

10 TOKEN DETAILS

11 ERROR404 FATHER ANALYTICS & TOP 10 TOKEN HOLDERS

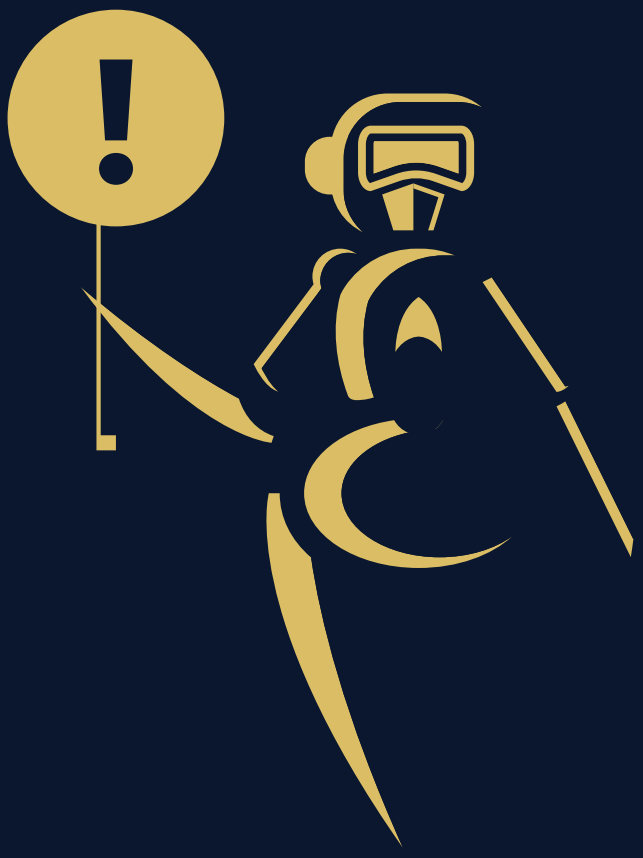
12 TECHNICAL DISCLAIMER

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 Error404 (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xc9E46c11c0810Ae41Aa2693be9323dDBe8AA0dBb

Network: Ethereum (ETH)

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 17/02/2024

WEBSITE DIAGNOSTIC

https://error404.meme/

 0-49

 50-89

 90-100

Performance

Accessibility

Best Practices

SEO

Progressive Web App

Socials

 Twitter

https://twitter.com/error404erc404

 Telegram

https://t.me/error404meme

AUDIT OVERVIEW

68

Security Score
HIGH RISK
Audit FAIL

93

Static Scan
Automatic scanning for common vulnerabilities

70

ERC Scan
Automatic checks for ERC's conformance

2	High
0	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 addresses from transactions

- Contract owner can change max tx amount (without threshold)

  Note that setting the value too low may prevent users from making purchase transactions

```
function setLimit(uint256 _buylimit, uint256 _selllimit) public onlyOwner{
    buyLimit = _buylimit;
    sellLimit = _selllimit;
}
```

- Contract owner can change name and symbol

  If the owner of a crypto token can change the name and symbol anytime, it introduces several risks and concerns for investors and users. Here are some potential issues: Confusion and Deception, Loss of Trust, Marketplace Listing Issues, Regulatory Scrutiny, Brand Recognition,Security Concerns

```
function setNameSymbol(
    string memory _name,
    string memory _symbol
) public onlyOwner {
    _setNameSymbol(_name, _symbol);
}
```

- Contract owner can set token URI

```
function setTokenURI(string memory _tokenURI) public onlyOwner {
    baseTokenURI = _tokenURI;
}
```

- Contract owner can whitelist address and remove it from limitations

```
function setWhitelist(address target, bool state) public onlyOwner {
    whitelist[target] = state;
}
```

- Contract owner has the capability to initiate or terminate transaction limitations

```
function startApplyingLimit() external onlyOwner{
    applyTxLimit = true;
}

function stopApplyingLimit() external onlyOwner{
    applyTxLimit = false;
}
```

Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public virtual onlyOwner {
    if (newOwner == address(0)) {
        revert OwnableInvalidOwner(address(0));
    }
    _transferOwnership(newOwner);
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {
    _transferOwnership(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 2 HIGH issues during the first review.

TOKEN DETAILS

Details

Buy fees:	N/A
Sell fees:	N/A
Max TX:	100
Max Sell:	100

Honeypot Risk

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

Rug Pull Risk

Liquidity:	N/A
Holders:	Clean

ERROR404 TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

Rank	Address	Quantity	Percentage	Analytics

There are no matching entries

Please try again later

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.