



## SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Gold  
\$GLD

05/03/2024

# TOKEN OVERVIEW

---

## Fees

- Buy fees: 2%
- Sell fees: 2%

## Fees privileges

- Can't change / set fees

## Ownership

- Owned

## Minting

- No mint function

## Max Tx Amount / Max Wallet Amount

- Can't change max tx amount (without threshold)

## Blacklist

- Blacklist function not detected

## Other privileges

- Can't exclude / include from tx limitations
  - The current implementation of transaction limitations and wallet restrictions is flawed, making it impossible to sell the token
-

# 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 GLD 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 Gold (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0xE277c0e1954f07D3ee8e3CA6D29f1FF73f0a1935

Network: Ethereum (sepolia TESTNET)

This report presents the findings of the security assessment of Customer's smart contract and its code review conducted on 05/03/2024



# WEBSITE DIAGNOSTIC

<https://www.mobamacoin.com/>



0-49



50-89



90-100



Performance



Accessibility



Best Practices



SEO



Progressive  
Web App

## Socials



Twitter

N/A



Telegram

N/A

# AUDIT OVERVIEW



**Security Score**



**Static Scan**

Automatic scanning for common vulnerabilities



**ERC Scan**

Automatic checks for ERC's conformance

**Audit FAIL**

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 addresses from transactions

- Contract owner can't exclude addresses from tax

We recommend implementing a function to exempt wallets from taxation. This is crucial in various situations where it becomes necessary to exclude an address from taxation, such as when dealing with the Pinksale Factory address for a launch on the PinkSale launchpad, among other scenarios.

- The current implementation of transaction limitations and wallet restrictions is flawed, making it impossible to sell the token

It is impossible to sell the token unless transaction and wallet limitations are disabled. However, if the sale is carried out from the owner's wallet, it functions properly.

Please check line 87-88 from File 1 of 7 : Token.sol

```
    .
    .
    .
    require(value <= _maxTx, "Transfer amount must be less than 47000000");
    require((value + _balances[to]) <= _maxWallet, "Wallet balance cannot exceed 47000000");
    .
    .
    .
```

- Contract owner can change `_taxWallet` address

Current value:

`_taxWallet: 0x2Cb4A426D7C6BB01651b0AbE216161aB2c8eca9D`

```
function changeTaxWallet(address wallet) external onlyOwner {
    _excluded[_taxWallet] = false;
    _excluded[wallet] = true;
    _taxWallet = wallet;
}
```

- Contract owner can set tx and wallet limitations to `totalSupply` value

```
function removeMaxLimits() external onlyOwner {
    _maxWallet = totalSupply();
    _maxTx = totalSupply();
    emit LimitsRemoved();
}
```

## ● Contract owner has ability to retrieve any token held by the contract

Native tokens NOT excluded

```
function withdrawETH() external {
    require(address(this).balance > 0, "Contract balance must be greater than zero");
    uint256 balance = address(this).balance;
    emit WithdrawETH(_taxWallet, balance);
    payable(_taxWallet).transfer(balance);
}

function withdrawTokens(address tokenAddress, uint256 amount) external returns (bool success) {
    if(amount == 0) {
        amount = IERC20(tokenAddress).balanceOf(address(this));
    }
    emit WithdrawToken(tokenAddress, _taxWallet, amount);
    return IERC20(tokenAddress).transfer(_taxWallet, amount);
}
```

## ● Contract owner can transfer ownership

```
function transferOwnership(address newOwner) public override onlyOwner {
    _excluded[owner()] = false;
    _excluded[newOwner] = true;
    super.transferOwnership(newOwner);
}

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

function _transferOwnership(address newOwner) internal virtual {
    address oldOwner = _owner;
    _owner = newOwner;
    emit OwnershipTransferred(oldOwner, newOwner);
}
```

## ● Contract owner can renounce ownership

```
function renounceOwnership() public override onlyOwner {
    _excluded[owner()] = false;
    super.renounceOwnership();
}

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 1 HIGH issues during the first review.

# TOKEN DETAILS

## Details

Buy fees:	2
Sell fees:	2
Max TX:	47,000,000
Max Wallet:	47,000,000

## Honeypot Risk

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

## Rug Pull Risk

Liquidity:	N/A
Holders:	Clean



# GLD 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.

