



SMART CONTRACT CODE REVIEW AND SECURITY ANALYSIS REPORT



Gold Shiba
\$GOSH

22/02/2022

TABLE OF CONTENTS

- 1 DISCLAIMER
- 2 INTRODUCTION
- 3-4 AUDIT OVERVIEW
- 5-7 OWNER PRIVILEGES
- 8 CONCLUSION AND ANALYSIS
- 9 TOKEN DETAILS
- 10 GOLD SHIBA TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS
- 11 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 Shiba (Customer) to conduct a Smart Contract Code Review and Security Analysis.

0x460e6C8F2Ddb06Db3BcBB24C67A75a59Dd949485

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 22/02/2022



AUDIT OVERVIEW



Security Score



Static Scan
Automatic scanning for common vulnerabilities



ERC Scan
Automatic checks for ERC's conformance

0 **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	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 exclude an address from transactions.

Contract owner can't mint tokens after initial contract deploy

Contract owner can exclude/include wallet from fees

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

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

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 already 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 max buy and sell tx amount

```
function setMaxBuyTxAmount(uint256 maxBuyTxAmount) external onlyOwner() {
    require(maxBuyTxAmount > 0, "transaction amount must be greater than zero");
    _maxBuyTxAmount = maxBuyTxAmount * (10**9);
}

function setMaxSellTxAmount(uint256 maxSellTxAmount) external onlyOwner() {
    require(maxSellTxAmount > 0, "transaction amount must be greater than zero");
    _maxSellTxAmount = maxSellTxAmount * (10**9);
}
```

Contract owner can change the fees up to 100%

```
function setAllFees(uint256 taxFee, uint256 liquidityFee, uint256 marketingFee, uint256 buyBackFee) private {
    _taxFee = taxFee;
    _liquidityFee = liquidityFee;
    _buyBackFee = buyBackFee;
    _marketingFee = marketingFee;
}

function setFees(uint256 taxFee, uint256 liquidityFee, uint256 marketingFee, uint256 buyBackFee)
    external onlyOwner() {
    setAllFees(taxFee, liquidityFee, marketingFee, buyBackFee);
    _previousTaxFee = taxFee;
    _previousLiquidityFee = liquidityFee;
    _previousBuyBackFee = buyBackFee;
    _previousMarketingFee = marketingFee;
}

function setSaleFees(uint256 taxFee, uint256 liquidityFee, uint256 marketingFee, uint256 buyBackFee)
    external onlyOwner() {
    _saleTaxFee = taxFee;
    _saleLiquidityFee = liquidityFee;
    _saleBuyBackFee = buyBackFee;
    _saleMarketingFee = marketingFee;
}
```

Contract owner can change marketingWallet address

Current values:

marketingWallet : 0x0bc13e2e5fde988e4312c5ff785790b62e3e0466

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

Contract owner can change buyback settings

```
function setBuyBackEnabled(bool _enabled) public onlyOwner {
    buyBackEnabled = _enabled;
    emit BuyBackEnabledUpdated(_enabled);
}

function SetBuyBackUpperLimit(uint256 _newAmount) external onlyOwner {
    buyBackUpperLimit = _newAmount;
}

function SetBuybackDivisor(uint256 divisor) external onlyOwner {
    buyBackDivisor = divisor;
}
```

Contract owner can change swap settings

```
function setSwapAndLiquifyEnabled(bool _enabled) public onlyOwner {  
    swapAndLiquifyEnabled = _enabled;  
    emit SwapAndLiquifyEnabledUpdated(_enabled);  
}
```

Contract owner can renounce ownership

```
function renounceOwnership() public virtual onlyOwner {  
    emit OwnershipTransferred(_owner, address(0));  
    _owner = 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");  
    emit OwnershipTransferred(_owner, newOwner);  
    _owner = newOwner;  
}
```



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 no issue during the first review.

TOKEN DETAILS

Details

Buy fees:	9%
Sell fees:	16%
Max TX:	1,000,000,000,000
Max Sell:	1,000,000,000,000

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



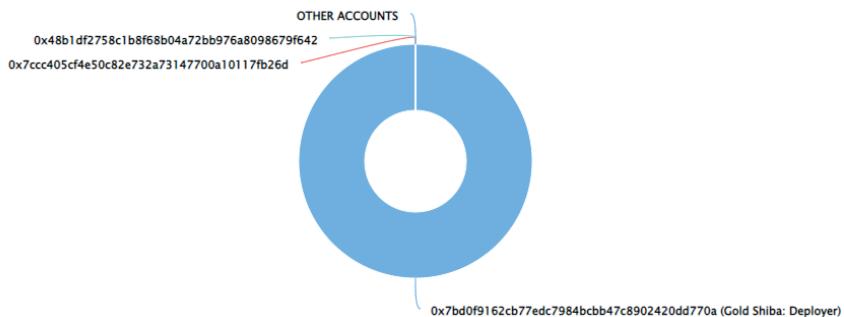
GOLD SHIBA TOKEN ANALYTICS & TOP 10 TOKEN HOLDERS

The top 10 holders collectively own 100.00% (999,990,750,000.00 Tokens) of Gold Shiba

Token Total Supply: 1,000,000,000,000.00 Token | Total Token Holders: 47

Gold Shiba Top 10 Token Holders

Source: BscScan.com



(A total of 999,990,750,000.00 tokens held by the top 10 accounts from the total supply of 1,000,000,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	Gold Shiba: Deployer	999,988,500,000	99.9989%
2	0x81100c5f57397d8de1469c2283b3f65c24eb0a07	250,000	0.0000%
3	0x43f9f7d8cc080ecaf68ba9276ffcaffa68373e2f	250,000	0.0000%
4	0xd31712192dfdf0e60f338e6c7566e81982f2c801	250,000	0.0000%
5	0x37e580829b307ef326169282eaa9b4d2559ca8af	250,000	0.0000%
6	0xca30b03b5737ffd55b58fb755bda1a66087a3b8d	250,000	0.0000%
7	0x37583703c5e315597d57324161bc244ebb8c399d	250,000	0.0000%
8	0x621f34b6215d09442b86a001ae9384bf0c367bff	250,000	0.0000%
9	0x7ccc405cf4e50c82e732a73147700a10117fb26d	250,000	0.0000%
10	0x48b1df2758c1b8f68b04a72bb976a8098679f642	250,000	0.0000%

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.

